REPORT NO. FAA-RD-78-29, IV Book 1.

STRUCTURAL AREA INSPECTION FREQUENCY EVALUATION (SAIFE)

Volume IV. Software Documentation and User's Manual

Book 1. Initial Program

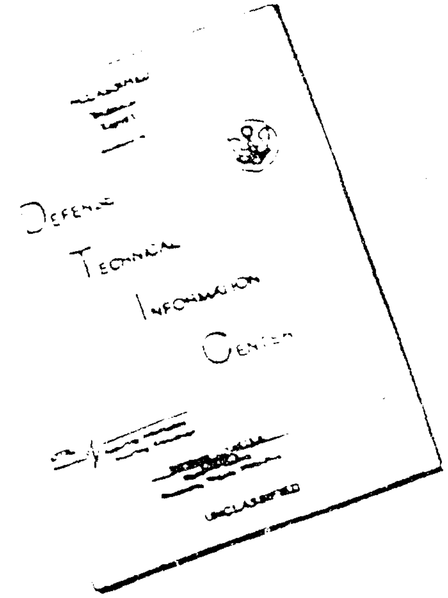
Carter J. Dinkeloo
Martin S. Moran

APRIL 1978
FINAL REPORT

Document is available to the U.S. public through
the National Technical Information Service,
Springfield, Virginia 22161.

Prepared for

U.S. DEPARTMENT OF TRANSPORTATION
FEDERAL AVIATION ADMINISTRATION
Systems Research & Development Service
Washington, D.C. 20590


Defense
Technical
Information
Center
UNCLASSIFIED

NOTICE

This document is disseminated under the sponsorship of the Department of Transportation in the interest of information exchange. The United States Government assumes no liability for its contents or use thereof.

Technical Report Documentation Page

1. Report No.
FAA-RD-78-29, IV, Book 1

2. Government Accession No.

3. Recipient's Catalog No.

4. Title and Subtitle
STRUCTURAL AREA INSPECTION FREQUENCY EVALUATION
Volume IV. Software Documentation and User's Manual. Book 1 - Initial Program

5. Report Date
April 1978

6. Performing Organization Code

8. Performing Organization Report No.

7. Author(s)
Carter J. Dinkeloo, Martin S. Moran

9. Performing Organization Name and Address
Technology Incorporated
P.O. Box 3036, Overlook Branch
Dayton, Ohio 45431

10. Work Unit No. (TRAIS)

11. Contract or Grant No.
DOT-FA74WA-3493

12. Sponsoring Agency Name and Address
Department of Transportation
Federal Aviation Administration
Systems Research and Development Service
Washington, D.C. 20590

13. Type of Report and Period Covered
Final Report

14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract

To assist in the evaluation of proposed structural inspection programs for commercial jet transport aircraft, a logic was developed to simulate structural defects, failures, and inspections. This logic was incorporated in a computer program entitled Structural Area Inspection Frequency Evaluation (SAIFE). With the objective of quantifying the evaluation process currently used to establish and modify inspection intervals, SAIFE accounts for the following factors: (1) aircraft design analysis; (2) fatigue testing; (3) production, service, and corrosion defects; (4) probability of crack or corrosion detection; and (5) aircraft modification economics. As a five-volume document, this report covers the initial contract effort plus a subsequent parametric analysis as follows: Volume I (entitled Executive Summary) presents the SAIFE logic and documents the methodology for the decision-making processes in the simulation logic. Volume II (entitled Description of Simulation Logic) details the SAIFE simulation logic, presents the background data for the analytical functions and decision-making processes, and includes data for a typical simulation problem. Volume III (entitled Demonstration Input, Inspection Survey, and MRR Data) presents data tabulations derived from historical trends and design input data for a SAIFE demonstration problem. As the user's manual for the SAIFE computer program, Volume IV contains detailed computer logic flow diagrams and a complete listing of the program which is written in SIMSCRIPT II.5. Volume V (entitled Results of Model Demonstration) presents the results of the program application to a hypothetical aircraft and compares these results with the service experience of operational aircraft.

17. Key Words

Aircraft experience simulation; aircraft inspection intervals; aircraft hazard rate and reliability; production, service, and corrosion defects; crack or corrosion detection probability; crack growth.

18. Distribution Statement
Document is available to the public through the National Technical Information Service, Springfield, Virginia 22151.

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
UNCLASSIFIED	UNCLASSIFIED	209	

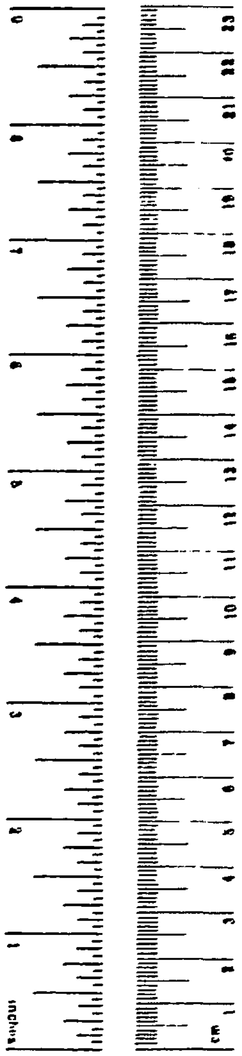
Reproduction of completed page authorized

METRIC CONVERSION FACTORS

Approximate Conversions to Metric Measures

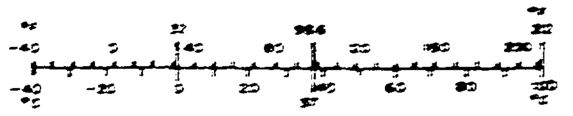
Symbol	When You Know	Multiply by	To Find	Symbol
		LENGTH		
in	inches	*2.5	centimeters	cm
ft	feet	30	centimeters	cm
yd	yards	0.9	meters	m
mi	miles	1.6	kilometers	km
		AREA		
in^2	square inches	6.5	square centimeters	cm^2
ft^2	square feet	0.09	square meters	m^2
yd^2	square yards	0.8	square meters	m^2
mi^2	square miles	2.6	square kilometers	km^2
	acres	0.4	hectares	ha
		MASS (weight)		
oz	ounces	28	grams	g
lb	pounds	0.45	kilograms	kg
	short tons (2000 lb)	0.9	tonnes	t
		VOLUME		
tsp	teaspoons	5	milliliters	ml
Tbsp	tablespoons	15	milliliters	ml
fl oz	fluid ounces	30	milliliters	ml
c	cups	0.24	liters	l
pt	pints	0.47	liters	l
qt	quarts	0.95	liters	l
gal	gallons	3.8	liters	l
ft^3	cubic feet	0.03	cubic meters	m^3
yd^3	cubic yards	0.76	cubic meters	m^3
		TEMPERATURE (exact)		
°F	Fahrenheit temperature	5/9 (after subtracting 32)	Celsius temperature	°C

*1 in = 2.54 (exactly). For other exact conversions and more detailed tables, see NBS Misc. Publ. 286, Units of Weights and Measures, Price $2.25, SD Catalog No. C13.10:286.



Approximate Conversions from Metric Measures

Symbol	When You Know	Multiply by	To Find	Symbol
		LENGTH		
mm	millimeters	0.04	inches	in
cm	centimeters	0.4	inches	in
m	meters	3.3	feet	ft
m	meters	1.1	yards	yd
km	kilometers	0.6	miles	mi
		AREA		
cm^2	square centimeters	0.16	square inches	in^2
m^2	square meters	1.2	square yards	yd^2
km^2	square kilometers	0.4	square miles	mi^2
ha	hectares (10,000 m^2)	2.5	acres	
		MASS (weight)		
g	grams	0.035	ounces	oz
kg	kilograms	2.2	pounds	lb
t	tonnes (1000 kg)	1.1	short tons	
		VOLUME		
ml	milliliters	0.03	fluid ounces	fl oz
l	liters	2.1	pints	pt
l	liters	1.06	quarts	qt
l	liters	0.26	gallons	gal
m^3	cubic meters	35	cubic feet	ft^3
m^3	cubic meters	1.3	cubic yards	yd^3
		TEMPERATURE (exact)		
°C	Celsius temperature	9/5 (then add 32)	Fahrenheit temperature	°F



PREFACE

Technology Incorporated prepared this fourth volume of a five-volume report to document the simulation logic for the Structural Area Inspection Frequency Evaluation (SAIFE) in accordance with Article II, paragraph B of Contract DOT-FA74WA-3493. (Volume IV along with Volume V completes the requirements of Phase III of the contract.) The effort is sponsored by the Aircraft Safety and Noise Abatement Division, Systems Research and Development Service of the Federal Aviation Administration.

The principal Technology Incorporated personnel engaged on this program were Mr. Carter J. Dinkeloo, project engineer, who served as principal investigator; Mr. Martin S. Moran, research engineer, who developed the model for the SAIFE computer program; and Mr. Ronald I. Rockafellow, program manager.

The contract monitors for the FAA were Messrs. Herbert Spicer and Charles Troha of the Aircraft Safety and Noise Abatement Division. The technical monitor was Mr. Arnold E. Anderjaska of the Flight Standards Division.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

I. INTRODUCTION

It is the mutual goal of the FAA, airframe manufacturers, and air carriers to constantly improve the structural integrity and inspection efficiency of civil aircraft. The good safety record of U.S. air carriers indicates that the current process of establishing and modifying structural inspection programs has been successful. However, with the increasing size and complexity of second- and third-generation transport aircraft, there is a need to quantify more precisely the present, subjective evaluation process which relies heavily on reliability analyses of the new design and on operational experience of similar aircraft.

Because of the extreme complexity of the evaluation process, a computer simulation of all critical aircraft service life aspects was judged the most rational means for quantifying the process more exactly. As a five-volume document, this report documents the resultant Structural Area Inspection Frequency Evaluation (SAIFE) simulation logic. SAIFE accounts for the following factors: (1) aircraft design analysis; (2) component and full-scale fatigue testing; (3) production, service, and corrosion defects; (4) probability of crack or corrosion detection; and (5) aircraft modification econimics. It treats these factors in a logical sequence that realistically represents the procedure currently used to establish and modify inspection intervals. Figure 1 illustrates the data sources and analytical functions that are integrated into the SAIFE logic. SAIFE is designed to provide a repeatable method for evaluating proposed inspection programs. However, it is not intended to supplant the Maintenance Review Board or the air carrier use of the Standard Operations Specification - Aircraft Maintenance.

As Volume IV, this user's manual for the SAIFE program contains a system description, a program description, operating procedures, a sample input and output, and a source listing of the program. The detailed description of the program events and routines is presented in Appendix A, and the program source listing is contained in Appendix B which because of its voluminous computer-generated data is on the included microfiche.

The original computer model developed during the initial contract has been modified by the Engineering and Manufacturing Branch, Flight Standards National Field Office for the parametric study and is documented in Book 2 of this volume.

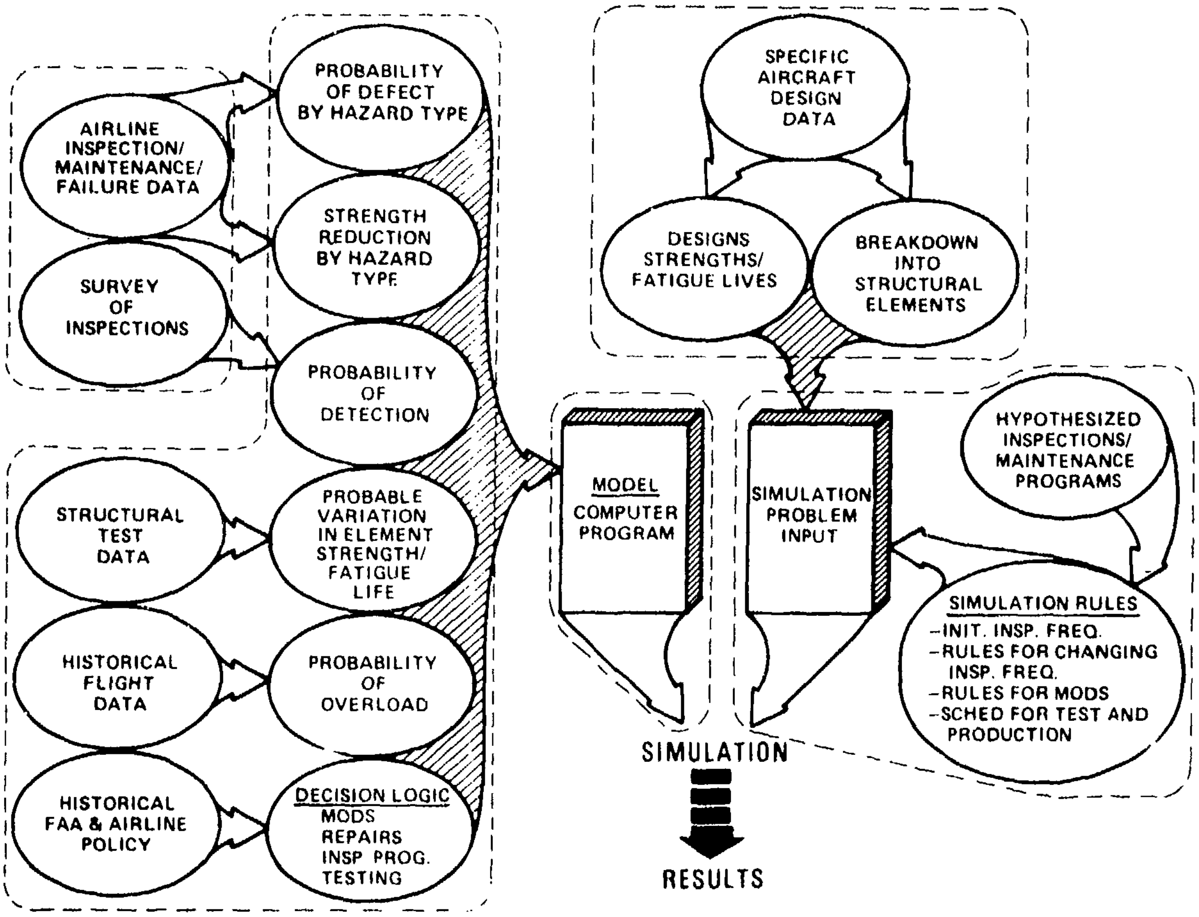

PROBABILITY OF DEFECT BY HAZARD TYPE
AIRLINE INSPECTION/ MAINTENANCE/ FAILURE DATA
STRENGTH REDUCTION BY HAZARD TYPE
SURVEY OF INSPECTIONS
PROBABILITY OF DETECTION
STRUCTURAL TEST DATA
PROBABLE VARIATION IN ELEMENT STRENGTH/ FATIGUE LIFE
HISTORICAL FLIGHT DATA
PROBABILITY OF OVERLOAD
HISTORICAL FAA & AIRLINE POLICY
DECISION LOGIC
MODS
REPAIRS
INSP PROG.
TESTING
SPECIFIC AIRCRAFT DESIGN DATA
DESIGNS STRENGTHS/ FATIGUE LIVES
BREAKDOWN INTO STRUCTURAL ELEMENTS
MODEL
COMPUTER PROGRAM
SIMULATION PROBLEM INPUT
HYPOTHESIZED INSPECTIONS/ MAINTENANCE PROGRAMS
SIMULATION RULES
-INIT. INSP. FREQ.
-RULES FOR CHANGING INSP. FREQ.
-RULES FOR MODS
-SCHED FOR TEST AND PRODUCTION
SIMULATION
RESULTS

Figure 1. Approach to SAIFE Simulation

II. SYSTEM DESCRIPTION

The eight blocks in Figure 2 represent the major aspects of the SAIFE simulation logic. Block 1 accepts input data for the aircraft fleet and for each structural element in the aircraft. After determining whether element modifications are required because of the fatigue test results in Block 2, Block 1 assigns a fatigue life to each element in each aircraft. Block 3 determines whether production, service, or corrosion defects will occur; if it is determined that such defects will occur, Block 3 predicts the times when they will occur. After comparing the flight loads with the strength of each element, Block 4 predicts the time to failure for each element. Block 5 conducts the periodic inspections of each element. If defects are detected, Block 6 repairs the element and assigns it a new fatigue life. However, if an existing defect is allowed to grow until element failure, Block 5 deletes the aircraft from the fleet. Depending on the magnitude of the detected defects, special inspections and increased inspection frequencies may be called for in Block 7 and modifications may be instituted in Block 8. When all the aircraft have been deleted from the fleet or retired from service, the simulation is complete.

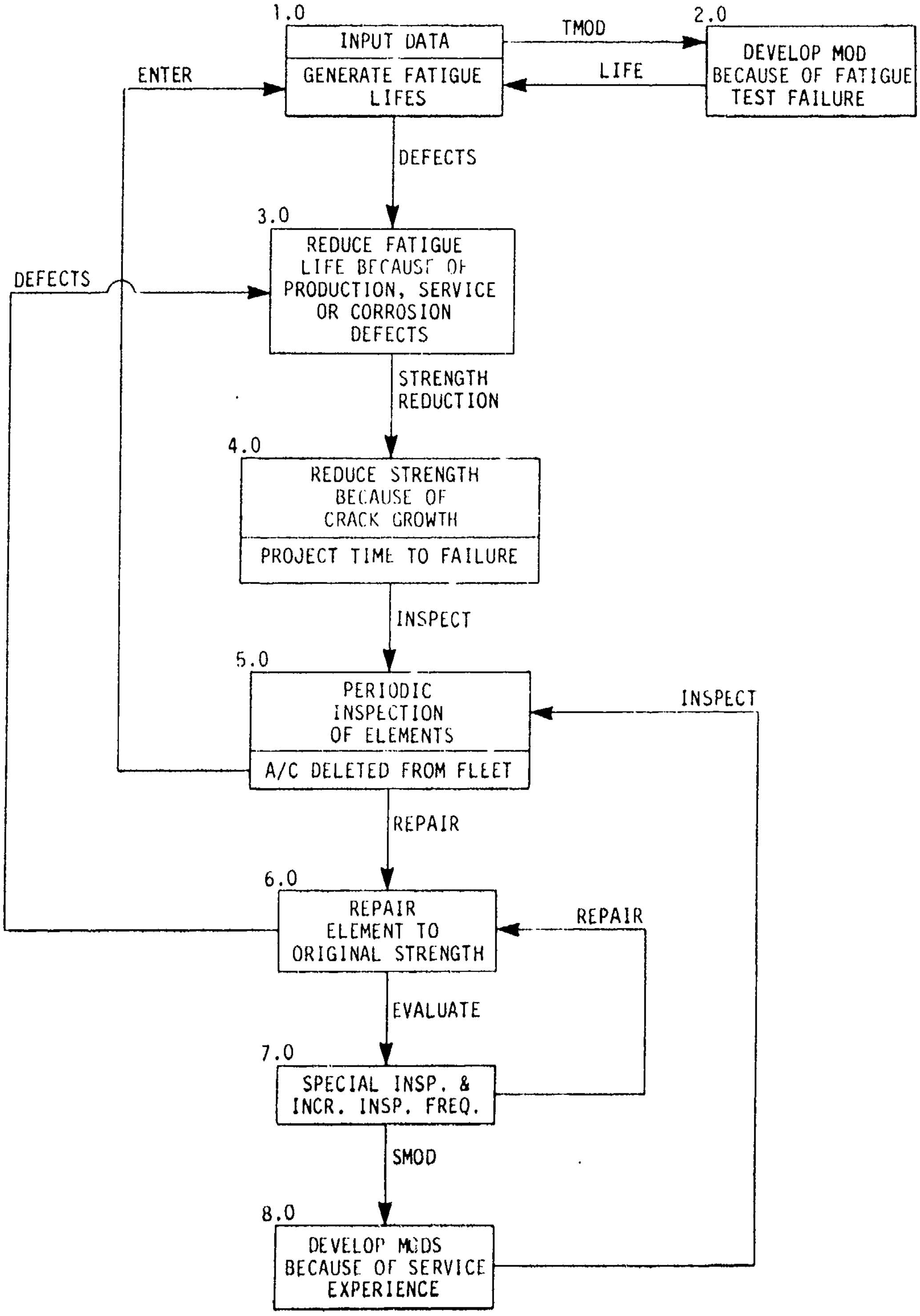

1.0
INPUT DATA
GENERATE FATIGUE LIFES
TMOD
LIFE
2.0
DEVELOP MOD BECAUSE OF FATIGUE TEST FAILURE
ENTER
DEFECTS
3.0
REDUCE FATIGUE LIFE BECAUSE OF PRODUCTION, SERVICE OR CORROSION DEFECTS
DEFECTS
STRENGTH REDUCTION
4.0
REDUCE STRENGTH BECAUSE OF CRACK GROWTH
PROJECT TIME TO FAILURE
INSPECT
5.0
PERIODIC INSPECTION OF ELEMENTS
A/C DELETED FROM FLEET
INSPECT
REPAIR
6.0
REPAIR ELEMENT TO ORIGINAL STRENGTH
REPAIR
EVALUATE
7.0
SPECIAL INSP. & INCR. INSP. FREQ.
SMOD
8.0
DEVELOP MODS BECAUSE OF SERVICE EXPERIENCE

Figure 2. Major Aspects of SAIFE Logic

III. GENERAL PROGRAM DESCRIPTION

The SAIFE program is a large, complex math model designed to simulate the structural performance of aircraft in a fleet of aircraft and the effectiveness of the inspection program for the aircraft fleet. The aircraft model included in the math model is divided into structurally significant elements and the inspection program for each element is defined. Structural defects are classified as follows: fatigue and corrosion which are wear-out and aging phenomena; production or design defects; and operational or maintenance damage. These defects and the inspection program are treated as probabilistic phenomena interacting over time. If the simulation is to proceed properly, the passage of simulated time must be controlled. This control can be accomplished by a user-designed algorithm or it can be done automatically with a "simulation clock" in one of the special purpose simulation languages. SIMSCRIPT II.5 ^{T} (trademark, Consolidated Analysis Centers Inc., Los Angeles, California), a computer language designed for discrete-event simulation applications, was chosen for the SAIFE simulation. SIMSCRIPT II.5 is a large language designed to facilitate the simulation of large, complex systems, and to reduce the total time spent in designing, programming, and testing simulation models.

Because of the extensive detail, the program events and routines are described in Appendix A.

IV. OPERATING PROCEDURES

1. Computer Requirements

The SAIFE demonstration output presented in Volume V of this report was run on an IBM Model 360-65 computer using the SIMSCRIPT II.5† compiler, release 8F. The execution time per element ranged between 12 and 20 cpu seconds, and the compilation time was 205 cpu seconds. None of the demonstration runs required more than 325K bytes of core storage.

The run time and the core storage requirements depend on the values of the input variables. The variable that probably has the greatest effect on these requirements is the actual average fatigue life. The shorter the life, the greater the requirement: A shorter actual average fatigue life increases crack initiations across the fleet. These additional crack initiations, in turn, increase the inspection process, the repair activities, and the modification and interval change decisions. All these additional events require more execution time and storage space for event notices. The actual average fatigue life is shortened when the parameters involved in the distribution of the ratio of the actual to the predicted average fatigue life are so changed that the ratio is decreased.

Other variables which affect the run time and core storage requirements are as follows: with the actual average fatigue life established for all aircraft in the fleet, the requirements are increased if the parameters involved in the Weibull distribution of actual fatigue lives for individual aircraft are so changed that some lives are shortened and more crack initiations are introduced. In addition, regardless of the fatigue lives, the run time and storage requirements are increased if the defect occurrence rates are increased or the initial inspection intervals are decreased.

Since the seeds of the random number streams are initialized to the same values at the start of each execution, the user can reproduce the output, if desired, and more easily identify effects because of changes in the input parameters. Table 1 lists the ten seeds used by the IBM 360-65 to generate the demonstration output presented in Volume V.

2. Input

The program input consists of three parts. The first part contains input variables which pertain to the aircraft type under consideration. These variables are input only once per simulation run and are constant from element to element. The second part is optional and is explained in Section IV,2.2. The third part contains input variables whose values are unique to each element. These variables must be input in their entirety for each element being simulated.

TABLE 1. INITIAL RANDOM ROOTS

```
SEED.V(1) = 2116429302
SEED.V(2) = 683743814
SEED.V(3) = 964393174
SEED.V(4) = 1217426631
SEED.V(5) = 618433579
SEED.V(6) = 1157240309
SEED.V(7) = 15726055
SEED.V(8) = 48108509
SEED.V(9) = 1797920909
SEED.V(10) = 477424540
```

2.1 Aircraft Data

The input variables which pertain to the aircraft type are listed and described below in the order in which they are read in by SAIFE.

MODEL(*) - This one-dimensional alpha array of size two identifies the aircraft type under consideration. The total length of this identification cannot exceed eight characters.

SIZE.OF.FLEET - This integer variable is the number of aircraft in the fleet being simulated. The output format requires that this variable does not exceed 99999.

USAGE.LIFE - This real variable is the service life in flight hours of the aircraft being simulated. All aircraft in the fleet must have the same service life. The output format requires that this variable does not exceed 9999999.

BEGIN.PRODUCTION - This real variable is the time in flight hours relative to the start of the simulation when the first aircraft enters service. This variable in conjunction with the input variable START.TEST enables the user to start the fatigue test of the element before, after, or at the same time the first aircraft enters service.

PRODUCTION.TIME - This real variable defines the initial aircraft production rate. It is the time in flight hours between aircraft entering service.

2.PRODUCTION.TIME - This real variable defines the second aircraft production rate. It is the time in flight hours between aircraft entering service.

PRCHG - This real variable is the simulation time when the second aircraft production rate takes effect. Note that this time is measured from the time that the first aircraft enters service and not from the start of the simulation.

START.TEST - This real variable is the time in flight hours relative to the start of the simulation when the fatigue test of an element is begun. If no fatigue test is to be conducted, this variable is set to the machine upper limit.

TEST.ACCEL.FACT - This real variable is the fatigue test acceleration factor, that is, the quotient of the equivalent flight hours divided by the fatigue test hours.

C.GROWTH.RATE - This real variable is the corrosion area growth rate in sq. inches per hour for the aircraft being considered. The growth rate for each element in the aircraft is modified by its associated CRR (corrosion resistance rating).

C7 - If a modification is developed because of a fatigue test failure, this real variable is the percentage (expressed as a decimal fraction) of the test life when the inspection frequency is increased.

C28 - This real variable is the percentage (expressed as a decimal fraction) reduction in the remaining fatigue life of an element when corrosion occurs in a stress concentration.

C29 - This real variable is the percentage (expressed as a decimal fraction) reduction in the remaining fatigue life of an element when corrosion occurs outside a stress concentration.

MU.R - This real variable is the mean of the log-normal distribution of the ratio of the actual average fatigue life to the predicted average fatigue life.

SIG.R - This real variable is the standard deviation of the log-normal distribution of the ratio of the actual average fatigue life to the predicted average fatigue life.

A - This real variable is the result of fitting an exponential curve to flight load exceedance data. $A\exp[BS_a]$ is the number of flight loads per hour which exceed the load level S_a.

B - This real variable is the result of fitting an exponential curve to flight load exceedance data. $A\exp[BS_a]$ is the number of flight loads per hour which exceed the load level S_a.

1ABCD(*) - This one-dimensional real array of size four contains the initial lengths in flight hours of the inspection intervals of the four levels of scheduled inspections. 1ABCD(1) corresponds to the A-level interval; 1ABCD(2) corresponds to the B-level interval; 1ABCD(3) corresponds to the C-level interval; and 1ABCD(4) corresponds to the D-level interval.

CABCD(*) - This one-dimensional real array of size four contains the inspection cost at each level of inspection. CABCD(1) corresponds to the A-level cost; CABCD(2) corresponds to the B-level cost; CABCD(3) corresponds to the C-level cost; and CABCD(4) corresponds to the D-level cost.

SAMPLING - This real variable is the percentage of the fleet to be sampled during a D-level inspection.

LONG.LIST - This alpha variable is read in as "YES" when the long list output option is desired; otherwise, it is read in as "NO."

PCCL - This real variable is the percentage (read in as a decimal fraction) of the element critical crack length at which a crack, which initiated internally, becomes external.

2.2 Long List Data

Occasionally in the standard output, elements will appear with unusually long fatigue cracks or early element failures. It is desirable to have a more complete service history of aircraft with these early element failures than that offered by the standard output. This service history is available through what is called the long list option. This output option is accessed by reading in alpha characters "YES" for the aircraft input variable LONG.LIST. After this input, the element description and identification numbers of the aircraft to be tracked are read in. The input variables for the long list option are listed and described below in the order in which they are read in by SAIFE.

NOE - This integer variable is the number of elements to be processed under the long list option.

ELID(*,*) - This two-dimensional alpha array of size four by NOE identifies each element to be processed. This identification must appear in the first sixteen columns of the data card and must be identical to the description read into the variable ELEMENT(*) described in Section IV,2.3.

NOAC(*) - This one-dimensional integer array of size NOE is the number of aircraft to be tracked for each corresponding element.

TLID(*,*) - This two-dimensional integer array of size NOE by NOAC(*) contains the identification numbers of the aircraft to be tracked for a particular element.

The above data are read in immediately following the aircraft input data only when the long list option is desired. Normally, the long list option will be used only after a standard run has indicated a problem area. Since the SAIFE program depends on many sequences of random numbers, all elements in the long list run must be in the same sequence as those in the first run. When the long list option is in effect, the standard output is suppressed.

2.3 Element Data

The input variables which are unique to each element and must be read in for each element are listed and described below in the order in which they are read in by SAIFE.

ELEMENT(*) - This one-dimensional alpha array of size four identifies the element being simulated. The total length of this identification cannot exceed sixteen characters.

PREDICTED.LIFE - This real variable is the average element fatigue life in flight hours predicted by analysis. If the actual average fatigue life is known, this variable can be entered as zero. The output format requires that this variable does not exceed 999999.

ACTUAL.AVG.FAT.LIFE - This real variable is the actual average element fatigue life in flight hours determined by fatigue test. If this value is not known before running the simulation, input zero and SAIFE will determine it statistically.

M1.MEAN - This real variable is the average slow crack propagation rate for the element being simulated.

M2.MEAN - This real variable is the average fast crack propagation rate for the element being simulated.

LGHT.TO.FAILURE - This real variable contains the element crack length which corresponds to zero residual strength or level-flight structural failure.

CRIT.CRK.LGT - This real variable is the element critical crack length, that is, the crack length when the crack propagation rate changes from slow to fast.

FSAF.LGT - This real variable is the element fail-safe crack length.

BIRTH.DEFECT.PROBABILITY - This real variable is the probability (expressed as a decimal fraction) that the element has a production defect when the aircraft enters service.

CRR - This integer variable is the element Corrosion Resistance Rating.

SDM.OCCURRENCE.RATE - This real variable is the service damage occurrence rate in occurrences per element per aircraft per flight hour.

LEAD.TIME - This real variable is the time in flight hours between the time when the decision is made to develop a modification and the time when the modification is ready for implementation.

T.FREQ.CHG - This real variable is the factor (expressed as a decimal fraction) by which inspection intervals are decreased because of a fatigue test failure.

S.FREQ.CHG - This real variable is the factor (expressed as a decimal fraction) by which inspection intervals are decreased because of unfavorable service experience.

FREQ.DECREASE - This real variable is the factor (expressed as a decimal fraction) by which inspection intervals are increased because of favorable service experience.

1.PROB - Given that there is fatigue crack initiation, this real variable is the probability that the crack initiates internally. This variable applies to all three allowable fatigue cracks.

C.PROB - Given that there is corrosion initiation, this real variable is the probability that the corrosion initiates internally.

INT.LVL.INSP - This alpha variable is the letter designation of the lowest internal inspection level.

EXT.LVL.INSP - This alpha variable is the letter designation of the lowest external inspection level.

A.REPAIR.COST - This real variable is the repair cost of a defect detected during an A-level inspection.

B.REPAIR.COST - This real variable is the repair cost of a defect detected during a B-level inspection.

C.REPAIR.COST - This real variable is the repair cost of a defect detected during a C-level inspection.

D.REPAIR.COST - This real variable is the repair cost of a defect detected during a D-level inspection.

MOD.TESTED - This alpha variable indicates whether or not a structural modification is to be fatigue tested. The two acceptable input values are YES or NO.

1ST.TOOLING - This real variable is the tooling cost in the development of the first structural modification.

AD.TOOLING - This real variable is the tooling cost in the development of any additional structural modifications.

1ST.MD.COST - This real variable is the installation cost of the first structural modification.

AD.MD.COST - This real variable is the installation cost of any additional structural modifications.

S.REPAIR.COST - This real variable is the repair cost of a defect detected during a special inspection.

LOCATED.IN.STRESS.CON - This real variable is the probability (expressed as a decimal fraction) that there is corrosion in a stress concentration.

1.CDM.OCCURRENCE.RATE - This real variable is the initial corrosion occurrence rate in occurrences per element per aircraft per flight hour.

2.CDM.OCCURRENCE.RATE - This real variable is the second corrosion occurrence rate in occurrences per element per aircraft per flight hour.

CDM.RATE.CHANGE - This real variable is the aircraft service time in flight hours when the second corrosion occurrence rate takes effect.

2.4 Format Specifications

Most of the input data are entered into SAIFE by the free-form read statement. The program has only three formatted read statements.

The aircraft type identification, the alpha array MODEL, is entered under the format specification 2A4. This identification must be contained in the first eight columns of the first card of the Aircraft Input Data. All subsequent data in this section can appear in any columns and on as many cards as desired. All input values must be separated from one another by at least one blank column and a value cannot be continued on the next card.

The element identification alpha array ELID is entered under the format specification 4A4. This identification must be contained in the first sixteen columns of the long list element data card. Subsequent data can appear in any column and on as many cards as desired. When a second element is to be identified, its description must again appear in the first sixteen columns of the data card.

The element identification alpha array ELEMENT is entered under the format specification 4A4. This identification must be contained in the first sixteen columns of the first card of each set of Element Input Data. As in the Aircraft Input Data, all subsequent data in this section can appear in any columns and on as many cards as desired.

Sample input data consisting of Aircraft Input Data and five sets of Element Input Data are illustrated in Figure 3. The aircraft type identification is HYBRID. The two cards immediately following the aircraft type card contain the Aircraft Input Data. The five elements shown in the sample are WSC-SWB-AFT-0000, WSC-SWB-AFT-0030, WSC-SWB-AFT-0060, WSC-SWB-AFT-0090, and WSC-SWB-AFT-0127. The Element Input Data for each element begins immediately after the element identification on the same card and terminates on the last card before the next element identification. The card following the last set of Element Input Data must contain EOD in the first four columns.

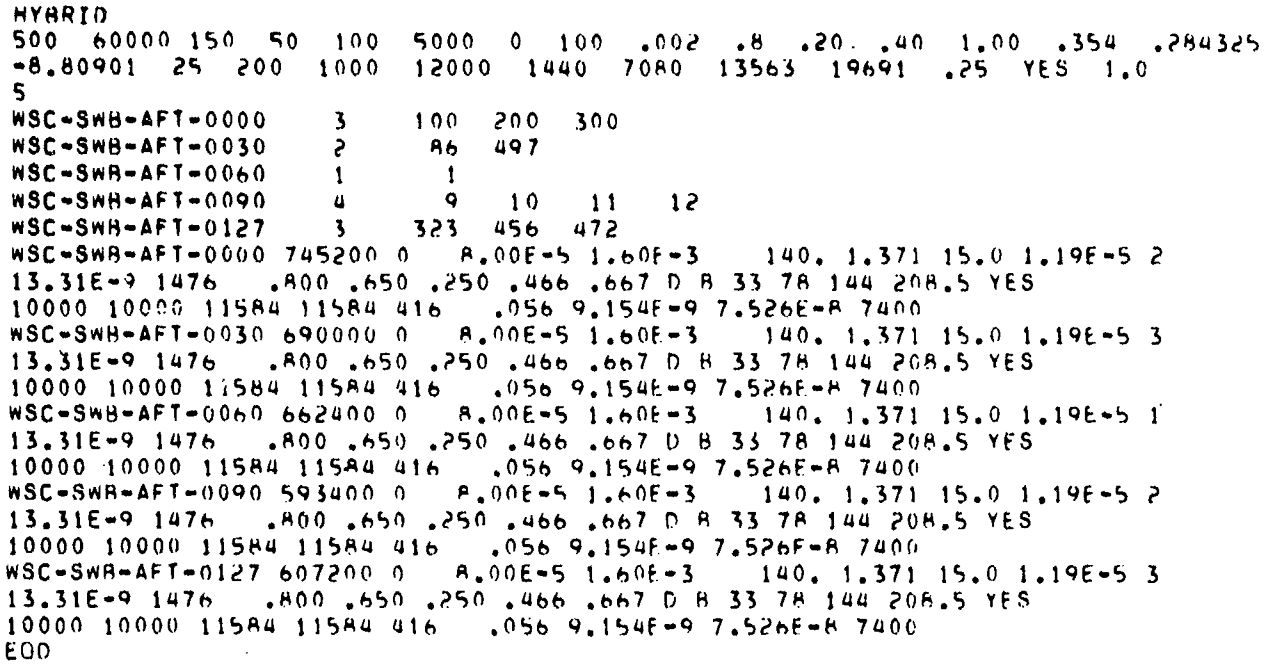
```
HYBRID
500  60000 150  50  100  5000  0  100  .002  .8  .20  .40  1.00  .354  .284325
-8.80901  25  200  1000  12000  1440  7080  13563  19691  .25  NO  1.0
WSC-SWB-AFT-0000 745200 0   8.00E-5 1.60E-3    140. 1.371 15.0 1.19E-5 2
13.31E-9 1476   .800 .650 .250 .466 .667 D B 33 78 144 208.5 YES
10000 10000 11584 11584 416   .056 9.154E-9 7.526E-8 7400
WSC-SWB-AFT-0030 690000 0   8.00E-5 1.60E-3    140. 1.371 15.0 1.19E-5 3
13.31E-9 1476   .800 .650 .250 .466 .667 D B 33 78 144 208.5 YES
10000 10000 11584 11584 416   .056 9.154E-9 7.526E-8 7400
WSC-SWB-AFT-0060 662400 0   8.00E-5 1.60E-3    140. 1.371 15.0 1.19E-5 1
13.31E-9 1476   .800 .650 .250 .466 .667 D B 33 78 144 208.5 YES
10000 10000 11584 11584 416   .056 9.154E-9 7.526E-8 7400
WSC-SWB-AFT-0090 593400 0   8.00E-5 1.60E-3    140. 1.371 15.0 1.19E-5 2
13.31E-9 1476   .800 .650 .250 .466 .667 D B 33 78 144 208.5 YES
10000 10000 11584 11584 416   .056 9.154E-9 7.526E-8 7400
WSC-SWB-AFT-0127 607200 0   8.00E-5 1.60E-3    140. 1.371 15.0 1.19E-5 3
13.31E-9 1476   .800 .650 .250 .466 .667 D B 33 78 144 208.5 YES
10000 10000 11584 11584 416   .056 9.154E-9 7.526E-8 7400
EOD
```

Figure 3. Sample Input Data to Produce Standard Output

Figure 4 illustrates the same data except that the variable LONG.LIST is now entered as "YES." The Long List Input data appears between the Aircraft Input Data and the Element Input Data. The Long List Input causes the program to track aircraft numbers 100, 200, and 300 for the element WSC-SWB-AFT-0000; aircraft numbers 86 and 497 for the element WSC-SWB-AFT-0030; aircraft number 1 for the element WSC-SWB-AFT-0060; aircraft numbers 9, 10, 11, and 12 for the element WSC-SWB-AFT-0090; and aircraft numbers 323, 456, and 472 for the element WSC-SWB-AFT-0127.

3. Output

Each element to be simulated by SAIFE is identified by three groups of alpha characters and one group of numeric characters. The alpha characters define the basic element type and general location on the aircraft, and the numeric characters define the specific location of the element by identifying the wing or fuselage station number. For example, an element identified as WNG-STR-CEN-396 would be a wing stringer located midway between the front and rear spars and centered at wing station 396.



```
HYBRID
500  60000 150  50  100  5000  0  100  .002  .8  .20  .40  1.00  .354  .284325
-8.80901  25  200  1000  12000  1440  7080  13563  19691  .25  YES  1.0
5
WSC-SWB-AFT-0000    3    100  200  300
WSC-SWB-AFT-0030    2     86  497
WSC-SWB-AFT-0060    1      1
WSC-SWB-AFT-0090    4      9   10   11   12
WSC-SWB-AFT-0127    3    323  456  472
WSC-SWB-AFT-0000 745200 0   8.00E-5 1.60E-3    140. 1.371 15.0 1.19E-5 2
13.31E-9 1476   .800 .650 .250 .466 .667 D B 33 78 144 208.5 YES
10000 10000 11584 11584 416   .056 9.154E-9 7.526E-8 7400
WSC-SWB-AFT-0030 690000 0   8.00E-5 1.60E-3    140. 1.371 15.0 1.19E-5 3
13.31E-9 1476   .800 .650 .250 .466 .667 D B 33 78 144 208.5 YES
10000 10000 11584 11584 416   .056 9.154E-9 7.526E-8 7400
WSC-SWB-AFT-0060 662400 0   8.00E-5 1.60E-3    140. 1.371 15.0 1.19E-5 1
13.31E-9 1476   .800 .650 .250 .466 .667 D B 33 78 144 208.5 YES
10000 10000 11584 11584 416   .056 9.154E-9 7.526E-8 7400
WSC-SWB-AFT-0090 593400 0   8.00E-5 1.60E-3    140. 1.371 15.0 1.19E-5 2
13.31E-9 1476   .800 .650 .250 .466 .667 D B 33 78 144 208.5 YES
10000 10000 11584 11584 416   .056 9.154E-9 7.526E-8 7400
WSC-SWB-AFT-0127 607200 0   8.00E-5 1.60E-3    140. 1.371 15.0 1.19E-5 3
13.31E-9 1476   .800 .650 .250 .466 .667 D B 33 78 144 208.5 YES
10000 10000 11584 11584 416   .056 9.154E-9 7.526E-8 7400
EOD
```

Figure 4. Sample Input Data to Produce Long List Output

The standard program output consists of two parts. The first part consists of the simulation results for each specific element. This part is printed for each set of Element Input Data. The second part consists of a summary of the first parts for an element type. In the example discussed above, WNG-STR-CEN identifies the element type. Whenever the program encounters a set of Element Input Data in which any single character of the three groups of alpha characters differs from those in the previous set of Element Input Data, a summary is printed.

A third output is available as an option. This long list option gives a more complete service history of certain selected aircraft and is discussed in Section IV,2.2. When the long list option is in effect, the standard output is suppressed.

3.1 Element Data

Figure 5 illustrates a sample output for the input shown in Figure 3. The aircraft type identification is the aircraft input array MODEL. The number of aircraft in the fleet is the aircraft input variable SIZE.OF.FLEET. The aircraft service life is the aircraft input variable USAGE.LIFE. The structural element identification is the element input array ELEMENT. The predicted average fatigue life is the element input variable PREDICTED.LIFE. The actual average fatigue life is the element input variable ACTUAL.AVG.FAT.LIFE.

```
                                 AIRCRAFT TYPE: HYBRID

               NUMBER OF AIRCRAFT IN FLEET:   500                    AIRCRAFT SERVICE LIFE:   60000 HOURS

                                    STRUCTURAL ELEMENT: WSC-SWB-AFT-0000

         PREDICTED AVERAGE FATIGUE LIFE: 745200 HOURS           ACTUAL AVERAGE FATIGUE LIFE:  670108 HOURS

                            NUMBER AND TIME TO INITIATION OF AIRCRAFT DEFECTS

                      FIRST CRACK         CORROSION         SERVICE DAMAGE        PRODUCTION DEFECTS
                      -----------         ---------         --------------        ------------------
                                                                                          0
 OCCURRENCES               2                   1                  2                     -----
 MIN(HRS)              32752               37720              32752                     -----
 MAX(HRS)              59789               37720              59789                     -----
 AVG(HRS)              46270               37720              46270

                        NUMBER AND LENGTH OF CRACKS DETECTED AT EACH LEVEL OF INSPECTION

                       A-LEVEL            B-LEVEL            C-LEVEL            D-LEVEL            SPECIAL
                       -------            -------            -------            -------            -------
 OCCURRENCES                 0                  1                  0                  0                  0
 MIN(IN)               0.                 2.22               0.                 0.                 0.
 MAX(IN)               0.                 2.22               0.                 0.                 0.
 AVG(IN)               0.                 2.22               0.                 0.                 0.

                     NUMBER AND AREA OF CORROSION DEFECTS DETECTED AT EACH LEVEL OF INSPECTION

                       A-LEVEL            B-LEVEL            C-LEVEL            D-LEVEL            SPECIAL
                       -------            -------            -------            -------            -------
 OCCURRENCES                 0                  0                  0                  0                  1
 MIN(SQ.IN)            0.                 0.                 0.                 0.                 35.47
 MAX(SQ.IN)            0.                 0.                 0.                 0.                 35.47
 AVG(SQ.IN)            0.                 0.                 0.                 0.                 35.47

 INSPECTION INTERVALS(HRS)

 INITIAL                    25                200               1000              12000
    2                       25                200               1250              15000
    3                       25                200               1563              18750
    4                       25                200               1953              23438
    5                       25                200                684               8203
    6                       25                200                854              10254
    7                       25                200               1068              12817

 NUMBER OF SPECIAL INSPECTIONS CONDUCTED:  1
 NUMBER OF STRUCTURAL MODIFICATIONS:  0
 FINAL ACTUAL AVERAGE MODIFIED FATIGUE LIFE:  670108 HOURS
 NUMBER OF AIRCRAFT MODIFIED IN SERVICE:      0

             STRUCTURAL FAILURES                             RESIDUAL STRENGTH EQUALS FAIL-SAFE STRENGTH
       AIRCRAFT NO.        FLT. HOURS                             AIRCRAFT NO.          FLT. HOURS
       ------------        ----------                             ------------          ----------
```

Figure 5. Sample Standard Output Produced by Input Shown in Figure 3

```
                                   AIRCRAFT TYPE: HYBRID

              NUMBER OF AIRCRAFT IN FLEET:   500                  AIRCRAFT SERVICE LIFE:   60000 HOURS

                                 STRUCTURAL ELEMENT: WSC-SWB-AFT-0030

           PREDICTED AVERAGE FATIGUE LIFE: 690000 HOURS          ACTUAL AVERAGE FATIGUE LIFE:  424261 HOURS

                            NUMBER AND TIME TO INITIATION OF AIRCRAFT DEFECTS

                       FIRST CRACK         CORROSION         SERVICE DAMAGE         PRODUCTION DEFECTS
                       -----------         ---------         --------------         ------------------
OCCURRENCES                  1                   1                   0                      0
MIN(HRS)                 51286               56541                   0                  -----
MAX(HRS)                 51286               56541                   0                  -----
AVG(HRS)                 51286               56541                   0                  -----

                          NUMBER AND LENGTH OF CRACKS DETECTED AT EACH LEVEL OF INSPECTION

                        A-LEVEL          B-LEVEL          C-LEVEL          D-LEVEL          SPECIAL
                        -------          -------          -------          -------          -------
OCCURRENCES                   0                0                0                0                0
MIN(IN)                 0.               0.               0.               0.               0.
MAX(IN)                 0.               0.               0.               0.               0.
AVG(IN)                 0.               0.               0.               0.               0.

                       NUMBER AND AREA OF CORROSION DEFECTS DETECTED AT EACH LEVEL OF INSPECTION

                        A-LEVEL          B-LEVEL          C-LEVEL          D-LEVEL          SPECIAL
                        -------          -------          -------          -------          -------
OCCURRENCES                   0                0                0                0                0
MIN(SQ.IN)              0.               0.               0.               0.               0.
MAX(SQ.IN)              0.               0.               0.               0.               0.
AVG(SQ.IN)              0.               0.               0.               0.               0.

INSPECTION INTERVALS(HRS)

INITIAL                  25              200             1000            12000
   2                     25              200             1250            15000
   3                     25              200             1563            18750
   4                     25              200             1953            23438

NUMBER OF SPECIAL INSPECTIONS CONDUCTED:  0
NUMBER OF STRUCTURAL MODIFICATIONS:  0
FINAL ACTUAL AVERAGE MODIFIED FATIGUE LIFE:  424261 HOURS
NUMBER OF AIRCRAFT MODIFIED IN SERVICE:      0

            STRUCTURAL FAILURES                          RESIDUAL STRENGTH EQUALS FAIL-SAFE STRENGTH
     AIRCRAFT NO.        FLT. HOURS                          AIRCRAFT NO.          FLT. HOURS
     ------------        ----------                          ------------          ----------
```

Figure 5 - Continued

```
                                   AIRCRAFT TYPE: HYBRID

          NUMBER OF AIRCRAFT IN FLEET:   500                 AIRCRAFT SERVICE LIFE:   60000 HOURS

                                STRUCTURAL ELEMENT: WSC-SWB-AFT-0060

    PREDICTED AVERAGE FATIGUE LIFE: 662400 HOURS              ACTUAL AVERAGE FATIGUE LIFE:  547824 HOURS

                            NUMBER AND TIME TO INITIATION OF AIRCRAFT DEFECTS

                    FIRST CRACK         CORROSION         SERVICE DAMAGE        PRODUCTION DEFECTS
                    -----------         ---------         --------------        ------------------
OCCURRENCES               2                  2                   0                       0
MIN(HRS)              41301              21125                   0                   -----
MAX(HRS)              58786              38060                   0                   -----
AVG(HRS)              50043              29592                   0                   -----

                         NUMBER AND LENGTH OF CRACKS DETECTED AT EACH LEVEL OF INSPECTION

                    A-LEVEL             B-LEVEL            C-LEVEL             D-LEVEL             SPECIAL
                    -------             -------            -------             -------             -------
OCCURRENCES               0                   1                  0                   0                   0
MIN(IN)                0.                1.12               0.                  0.                  0.
MAX(IN)                0.                1.12               0.                  0.                  0.
AVG(IN)                0.                1.12               0.                  0.                  0.

                      NUMBER AND AREA OF CORROSION DEFECTS DETECTED AT EACH LEVEL OF INSPECTION

                    A-LEVEL             B-LEVEL            C-LEVEL             D-LEVEL             SPECIAL
                    -------             -------            -------             -------             -------
OCCURRENCES               0                   1                  0                   1                   0
MIN(SQ.IN)             0.                6.61               0.                 17.63                0.
MAX(SQ.IN)             0.                6.61               0.                 17.63                0.
AVG(SQ.IN)             0.                6.61               0.                 17.63                0.

INSPECTION INTERVALS(HRS)

INITIAL                  25                 200               1000               12000
   2                     25                 200               1250               15000
   3                     25                 200               1563               18750
   4                     25                 200               1953               23438
   5                     25                 200                684                8203
   6                     25                 200                854               10254
   7                     25                 200               1068               12817
   8                     25                 200               1335               16022
   9                     25                 200               1669               20027

NUMBER OF SPECIAL INSPECTIONS CONDUCTED:  1
NUMBER OF STRUCTURAL MODIFICATIONS:  0
FINAL ACTUAL AVERAGE MODIFIED FATIGUE LIFE:  547824 HOURS
NUMBER OF AIRCRAFT MODIFIED IN SERVICE:      0

          STRUCTURAL FAILURES                              RESIDUAL STRENGTH EQUALS FAIL-SAFE STRENGTH
      AIRCRAFT NO.        FLT. HOURS                             AIRCRAFT NO.        FLT. HOURS
      ------------        ----------                             ------------        ----------
```

Figure 5 - Continued

```
                                        AIRCRAFT TYPE: HYBRID

              NUMBER OF AIRCRAFT IN FLEET:   500                    AIRCRAFT SERVICE LIFE:   60000 HOURS

                                   STRUCTURAL ELEMENT: WSC-SWB-AFT-0090

       PREDICTED AVERAGE FATIGUE LIFE: 593400 HOURS             ACTUAL AVERAGE FATIGUE LIFE:  539763 HOURS

                                NUMBER AND TIME TO INITIATION OF AIRCRAFT DEFECTS

                       FIRST CRACK         CORROSION          SERVICE DAMAGE        PRODUCTION DEFECTS
                       -----------         ---------          --------------        ------------------
OCCURRENCES                  6                  6                    3                      0
MIN(HRS)                  1603               5542                 1603                  -----
MAX(HRS)                 48952              56707                48952                  -----
AVG(HRS)                 33185              25377                23168                  -----

                          NUMBER AND LENGTH OF CRACKS DETECTED AT EACH LEVEL OF INSPECTION

                       A-LEVEL            B-LEVEL            C-LEVEL            D-LEVEL            SPECIAL
                       -------            -------            -------            -------            -------
OCCURRENCES                  0                  2                  2                  1                  0
MIN(IN)                   0.                .51                .49               1.05                0.
MAX(IN)                   0.                .65                .55               1.05                0.
AVG(IN)                   0.                .58                .52               1.05                0.

                      NUMBER AND AREA OF CORROSION DEFECTS DETECTED AT EACH LEVEL OF INSPECTION

                       A-LEVEL            B-LEVEL            C-LEVEL            D-LEVEL            SPECIAL
                       -------            -------            -------            -------            -------
OCCURRENCES                  0                  2                  0                  1                  0
MIN(SQ.IN)                0.               1.65               0.                17.84               0.
MAX(SQ.IN)                0.               2.08               0.                17.84               0.
AVG(SQ.IN)                0.               1.86               0.                17.84               0.

INSPECTION INTERVALS(HRS)

INITIAL                     25                200               1000              12000
   2                        25                200               1250              15000
   3                        25                200               1563              18750
   4                        25                200               1953              23438
   5                        25                200                684               8203
   6                        25                200                854              10254
   7                        25                200               1068              12817
   8                        25                200               1335              16022
   9                        25                200               1669              20027

NUMBER OF SRECIAL INSPECTIONS CONDUCTED:  1
NUMBER OF STRUCTURAL MODIFICATIONS:  0
FINAL ACTUAL AVERAGE MODIFIED FATIGUE LIFE:  539763 HOURS
NUMBER OF AIRCRAFT MODIFIED IN SERVICE:     0

           STRUCTURAL FAILURES                                   RESIDUAL STRENGTH EQUALS FAIL-SAFE STRENGTH
      AIRCRAFT NO.         FLT. HOURS                                  AIRCRAFT NO.        FLT. HOURS
      ------------         ----------                                  ------------        ----------
```

Figure 5 - Continued

```
                                         AIRCRAFT TYPE: HYBRID

              NUMBER OF AIRCRAFT IN FLEET:   500                    AIRCRAFT SERVICE LIFE:   60000 HOURS

                                   STRUCTURAL ELEMENT: WSC-SWB-AFT-0127

          PREDICTED AVERAGE FATIGUE LIFE: 607200 HOURS              ACTUAL AVERAGE FATIGUE LIFE:  976224 HOURS

                                NUMBER AND TIME TO INITIATION OF AIRCRAFT DEFECTS

                          FIRST CRACK         CORROSION         SERVICE DAMAGE         PRODUCTION DEFECTS
                          -----------         ---------         --------------         ------------------
OCCURRENCES                    1                  3                    0                        0
MIN(HRS)                     46529              25520                  0                      -----
MAX(HRS)                     46529              59353                  0                      -----
AVG(HRS)                     46529              43195                  0                      -----

                            NUMBER AND LENGTH OF CRACKS DETECTED AT EACH LEVEL OF INSPECTION

                          A-LEVEL           B-LEVEL           C-LEVEL           D-LEVEL           SPECIAL
                          -------           -------           -------           -------           -------
OCCURRENCES                  0                 1                 0                 0                 0
MIN(IN)                    0.                3.33              0.                0.                0.
MAX(IN)                    0.                3.33              0.                0.                0.
AVG(IN)                    0.                3.33              0.                0.                0.

                         NUMBER AND AREA OF CORROSION DEFECTS DETECTED AT EACH LEVEL OF INSPECTION

                          A-LEVEL           B-LEVEL           C-LEVEL           D-LEVEL           SPECIAL
                          -------           -------           -------           -------           -------
OCCURRENCES                  0                 1                 0                 1                 0
MIN(SQ.IN)                 0.                1.38              0.                2.96              0.
MAX(SQ.IN)                 0.                1.38              0.                2.96              0.
AVG(SQ.IN)                 0.                1.38              0.                2.96              0.

INSPECTION INTERVALS(HRS)

INITIAL                     25                200              1000             12000
   2                        25                200              1250             15000
   3                        25                200              1563             18750
   4                        25                200              1953             23438
   5                        25                200               684              8203
   6                        25                200               854             10254
   7                        25                200              1068             12817

NUMBER OF SPECIAL INSPECTIONS CONDUCTED:  1
NUMBER OF STRUCTURAL MODIFICATIONS:  0
FINAL ACTUAL AVERAGE MODIFIED FATIGUE LIFE:  976224 HOURS
NUMBER OF AIRCRAFT MODIFIED IN SERVICE:      0

            STRUCTURAL FAILURES                               RESIDUAL STRENGTH EQUALS FAIL-SAFE STRENGTH
       AIRCRAFT NO.       FLT. HOURS                                AIRCRAFT NO.       FLT. HOURS
       ------------       ----------                                ------------       ----------
```

Figure 5 - Continued

```
                              AIRCRAFT TYPE: HYBRID

        NUMBER OF AIRCRAFT IN FLEET:   500                 AIRCRAFT SERVICE LIFE:   60000 HOURS

                            SUMMARY OF STRUCTURAL ELEMENT: ASC-SAB-AFT

                        NUMBER AND TIME TO INITIATION OF AIRCRAFT DEFECTS

                FIRST CRACK        CORROSION        SERVICE DAMAGE      PRODUCTION DEFECTS
                -----------        ---------        --------------      ------------------
OCCURRENCES          12                13                  5                     0
MIN(HRS)           1603              5542               1603                 -----
MAX(HRS)          59780             59353              59780                 -----
AVG(HRS)          49706             33464              32490                 -----

                   NUMBER AND LENGTH OF CRACKS DETECTED AT EACH LEVEL OF INSPECTION

                A-LEVEL          B-LEVEL          C-LEVEL          D-LEVEL          SPECIAL
                -------          -------          -------          -------          -------
OCCURRENCES         0                5                2                1                0
MIN(IN)         0.                .51              .60             1.05              0.
MAX(IN)         0.               3.33              .55             1.05              0.
AVG(IN)         0.               1.57              .52             1.05              0.

               NUMBER AND AREA OF CORROSION DEFECTS DETECTED AT EACH LEVEL OF INSPECTION

                A-LEVEL          B-LEVEL          C-LEVEL          D-LEVEL          SPECIAL
                -------          -------          -------          -------          -------
OCCURRENCES         0                4                0                3                1
MIN(SQ.IN)      0.               1.38             0.               2.96            35.47
MAX(SQ.IN)      0.               6.61             0.              17.84            35.47
AVG(SQ.IN)      0.               2.93             0.              12.81            35.87

INSPECTION INTERVALS(HRS)
INITIAL             25              200             1000            12000
SHORTEST            25              200              684             8203
LONGEST             25              200             1953            23458

NUMBER OF SPECIAL INSPECTIONS CONDUCTED:   4
NUMBER OF STRUCTURAL MODIFICATIONS:   0
NUMBER OF AIRCRAFT MODIFIED IN SERVICE:    0

            STRUCTURAL FAILURES                             RESIDUAL STRENGTH EQUALS FAIL-SAFE STRENGTH
AIRCRAFT NO.     FLT. HOURS       STA. NO.                AIRCRAFT NO.      FLT. HOURS        STA. NO.
------------     ----------       --------                ------------      ----------        --------
```

Figure 5 - Concluded

The number of occurrences in the fleet and the times to initiation of the four types of aircraft defects considered by SAIFE are displayed next. Whenever there is a fatigue crack initiation and there are no other cracks in the element, a first crack is said to have occurred. A single element can experience more than one first crack in its lifetime by having a crack initiation after a repair. Similarly, there can be more than one occurrence of corrosion and service damage, although the program does not allow more than one corrosion or service damage defect to exist simultaneously in the same element.

Production defects are one-time occurrences unless there is a structural modification installed. These, too, can have production defects. The times to defect initiation are measured from the time when the aircraft enters service for the initial defects and from the time when the aircraft was last repaired for subsequent defects.

Next, the number and lengths of cracks detected in the fleet at each level of inspection are printed. These numbers include second and third crack detections. Following the crack detection output are the number and areas of corrosion defects detected in the fleet at each level of inspection. A history of the inspection interval changes is printed next. Each time that the aircraft service experience indicates that either an interval increase or an interval decrease is needed, the new interval values are printed. Although the number of interval changes allowed in the simulation is unlimited, the output array size limits the number printed to 30.

The number of fleet-wide special inspections performed s printed next. Each special inspection is always preceded by a decrease in inspection intervals. Defects detected during a special inspection can cause an additional decrease in inspection intervals. Next, the number of structural modifications developed is printed. This number includes modifications because of fatigue test failures or aircraft service experience. The final actual average fatigue life is printed next. If there have been no modifications, this number will be the same as that at the top of the page. If there have been modifications, this number is the actual average fatigue life of the most recently developed modification.

Shown next s the number of aircraft modified in service. If the only modification developed was due to a fatigue test failure, this number can be zero if the test life was such that retrofits were not reqired. If there were more than one modification requiring retrofits, this number can be greater than the size of the fleet. Finally, each time an aircraft experiences structural failure or its residual strength reaches its fail-safe strength, the aircraft number and the number of accrued flight hours are printed. The aircraft number is assigned by its relative time of entry into service. Aircraft No. 1 is the first aircraft to enter service.

3.2 Summary Data

The last section of Figure 5 illustrates a sample summary for the element type WSC-SWB-AFT. All the numbers represent a summary of all the specific elements of this type. Except for the shortest and longest inspection intervals, each number in the summary will appear in one of the specific element outputs. As indicated earlier, the number of interval changes allowed in the simulation is unlimited, and the number printed for a specific element is limited to 30. The shortest and longest intervals printed in the summary are determined from the unlimited number of changes occurring in the simulation.

3.3 Long List Data

Figure 6 illustrates a sample long list output for the input shown in Figure 4. For each element the long list headings contain the following: aircraft description, number of aircraft in the fleet, aircraft service life, element description, predicted average fatigue life of the element, actual average fatigue life of the element, and the initial inspection intervals. For each aircraft being tracked, the long list option causes selected information to be printed each time the program control passes to certain events and routines. These events and routines along with the information printed are the following:

Event ENTER.SERVICE - Prints aircraft identification number, number of hours from start of simulation, projected flignt hours until crack initiations, and the slow and fast crack growth rates.

Routine INSTALL.MODIFICATION - Prints aircraft identification number, flight time on aircraft, flight hours until crack initiations, and the slow and fast crack growth rates.

Event IN.SERVICE.DAMAGE - Prints aircraft identification number and flight time on aircraft.

Event CORROSION - Prints aircraft identification number, flight time on aircraft, revised slow and fast crack growth rates, revised times until crack initiations, and revised time until failure.

Event 1.STRENGTH.REDUCTION - Prints aircraft identification number, flight time on aircraft, and projected flight hours until element failure.

Event 2.STRENGTH.REDUCTION - Prints aircraft identification number, flight time on aircraft, and projected flight hours until element failure.

Event 3.STRENGTH.REDUCTION - Prints aircraft identification number, flight time on aircraft, and projected flight hours until element failure.

Event 1.ITE - Prints aircraft identification number, length of crack, and flight time on aircraft.

Event 2.ITE - Prints aircraft identification number, length of crack, and flight time on aircraft.

Event 3.ITE - Prints aircraft identification number, length of crack, and flight time on aircraft.

Event D.LEVEL.INSPECTION - Prints aircraft identification number and flight time on aircraft. If an inspection interval increase is implemented at this time, revised intervals are also printed.

Routine EXAMINE - For each defect found, prints size of defect, level of inspection, aircraft identification number, and flight time on aircraft.

Event REACH.FAIL.SAFE.LGT - Prints aircraft identification number and flight time on aircraft.

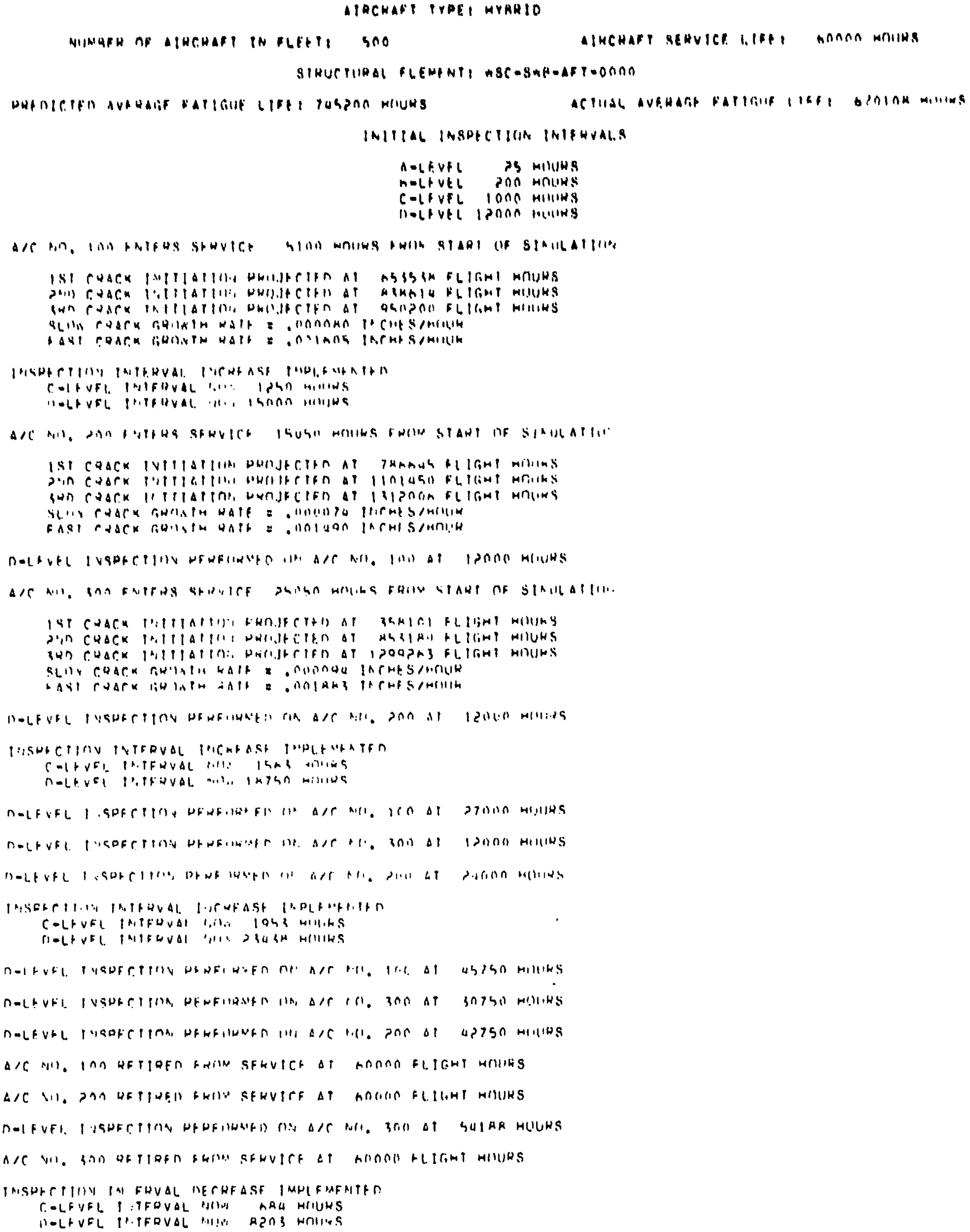
Event FAILURE - Prints aircraft identification number, flight time on aircraft, sum of crack lengths, and element residual strength.

Event RETIRE.FROM.SERVICE - Prints aircraft identification number and flight time on aircraft.

Event REPAIR - Prints aircraft identification number and projected times to crack initiations.

Event INCREASE.INSPECTION.FREQUENCY - Prints revised C-level and D-level inspection intervals.

Event IMMEDIATE.FLEET.INSPECTION - For each defect found, prints type of defect, size of defect, aircraft identification number, and flight time on aircraft.



```
                                    AIRCRAFT TYPE: HYBRID

          NUMBER OF AIRCRAFT IN FLEET:   500                   AIRCRAFT SERVICE LIFE:   60000 HOURS

                            STRUCTURAL ELEMENT: WSC-SWB-AFT-0000

PREDICTED AVERAGE FATIGUE LIFE: 745200 HOURS              ACTUAL AVERAGE FATIGUE LIFE:  670108 HOURS

                                   INITIAL INSPECTION INTERVALS

                                       A-LEVEL     25 HOURS
                                       B-LEVEL    200 HOURS
                                       C-LEVEL   1000 HOURS
                                       D-LEVEL  12000 HOURS

A/C NO. 100 ENTERS SERVICE    5100 HOURS FROM START OF SIMULATION

    1ST CRACK INITIATION PROJECTED AT   653558 FLIGHT HOURS
    2ND CRACK INITIATION PROJECTED AT   838614 FLIGHT HOURS
    3RD CRACK INITIATION PROJECTED AT   950200 FLIGHT HOURS
    SLOW CRACK GROWTH RATE = .000080 INCHES/HOUR
    FAST CRACK GROWTH RATE = .001805 INCHES/HOUR

INSPECTION INTERVAL INCREASE IMPLEMENTED
    C-LEVEL INTERVAL NOW   1250 HOURS
    D-LEVEL INTERVAL NOW  15000 HOURS

A/C NO. 200 ENTERS SERVICE   15050 HOURS FROM START OF SIMULATION

    1ST CRACK INITIATION PROJECTED AT   788845 FLIGHT HOURS
    2ND CRACK INITIATION PROJECTED AT  1101450 FLIGHT HOURS
    3RD CRACK INITIATION PROJECTED AT  1312006 FLIGHT HOURS
    SLOW CRACK GROWTH RATE = .000074 INCHES/HOUR
    FAST CRACK GROWTH RATE = .001490 INCHES/HOUR

D-LEVEL INSPECTION PERFORMED ON A/C NO. 100 AT  12000 HOURS

A/C NO. 300 ENTERS SERVICE   25050 HOURS FROM START OF SIMULATION

    1ST CRACK INITIATION PROJECTED AT   358161 FLIGHT HOURS
    2ND CRACK INITIATION PROJECTED AT   853180 FLIGHT HOURS
    3RD CRACK INITIATION PROJECTED AT  1299263 FLIGHT HOURS
    SLOW CRACK GROWTH RATE = .000094 INCHES/HOUR
    FAST CRACK GROWTH RATE = .001883 INCHES/HOUR

D-LEVEL INSPECTION PERFORMED ON A/C NO. 200 AT  12000 HOURS

INSPECTION INTERVAL INCREASE IMPLEMENTED
    C-LEVEL INTERVAL NOW   1563 HOURS
    D-LEVEL INTERVAL NOW  18750 HOURS

D-LEVEL INSPECTION PERFORMED ON A/C NO. 100 AT  27000 HOURS

D-LEVEL INSPECTION PERFORMED ON A/C NO. 300 AT  12000 HOURS

D-LEVEL INSPECTION PERFORMED ON A/C NO. 200 AT  24000 HOURS

INSPECTION INTERVAL INCREASE IMPLEMENTED
    C-LEVEL INTERVAL NOW   1953 HOURS
    D-LEVEL INTERVAL NOW  23438 HOURS

D-LEVEL INSPECTION PERFORMED ON A/C NO. 100 AT  45750 HOURS

D-LEVEL INSPECTION PERFORMED ON A/C NO. 300 AT  30750 HOURS

D-LEVEL INSPECTION PERFORMED ON A/C NO. 200 AT  42750 HOURS

A/C NO. 100 RETIRED FROM SERVICE AT  60000 FLIGHT HOURS

A/C NO. 200 RETIRED FROM SERVICE AT  60000 FLIGHT HOURS

D-LEVEL INSPECTION PERFORMED ON A/C NO. 300 AT  54188 HOURS

A/C NO. 300 RETIRED FROM SERVICE AT  60000 FLIGHT HOURS

INSPECTION INTERVAL DECREASE IMPLEMENTED
    C-LEVEL INTERVAL NOW    684 HOURS
    D-LEVEL INTERVAL NOW   8203 HOURS
```

a. Structural Element: WSC-SWB-AFT-0000

Figure 6. Sample Long List Output Produced by Input Shown in Figure 4

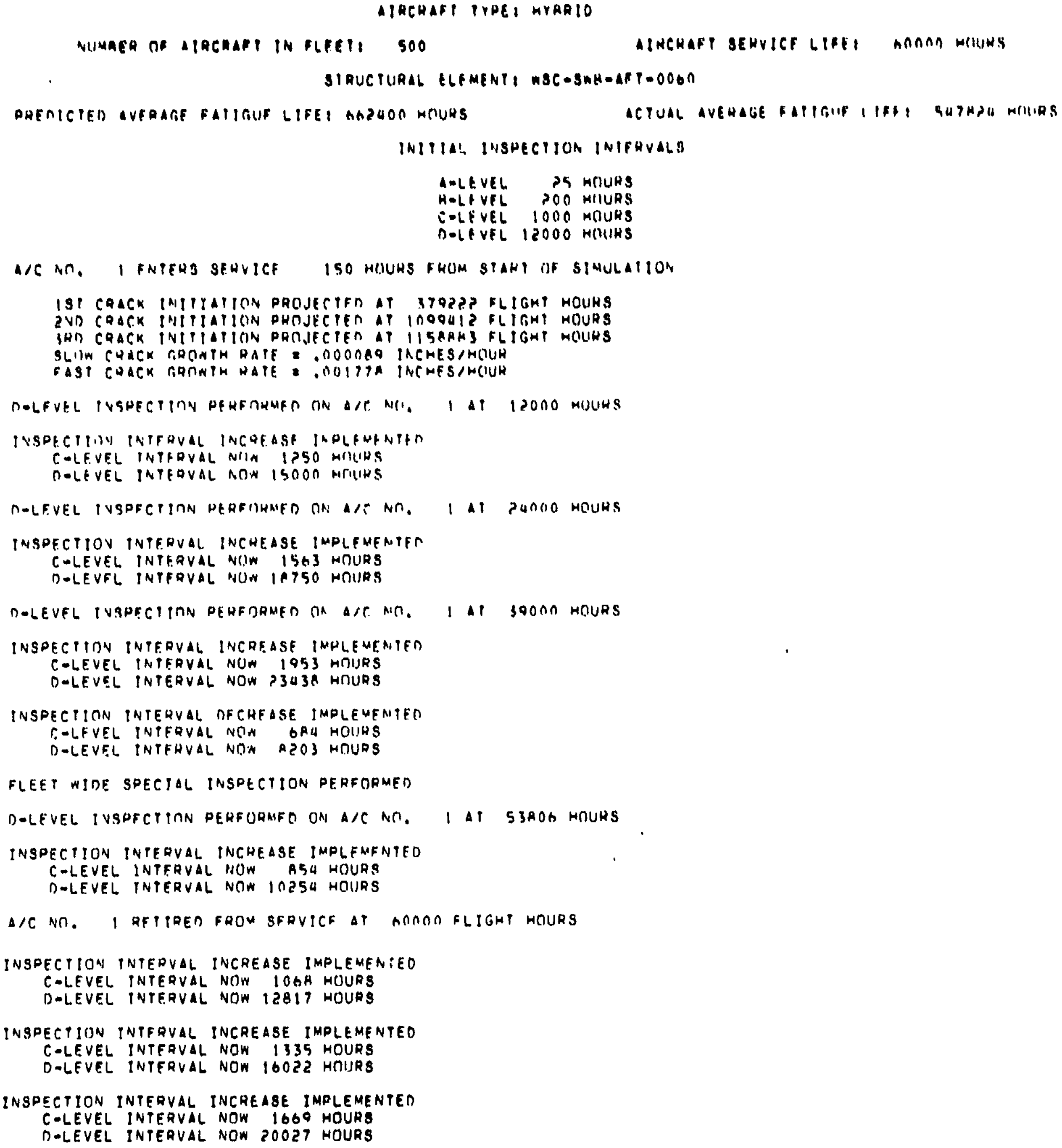
```
FLEET 5% - SPECIAL INSPECTION PERFORMED

INSPECTION INTERVAL INCREASE IMPLEMENTED
    C-LEVEL INTERVAL NOW   854 HOURS
    D-LEVEL INTERVAL NOW 10250 HOURS

INSPECTION INTERVAL INCREASE IMPLEMENTED
    C-LEVEL INTERVAL NOW  1068 HOURS
    D-LEVEL INTERVAL NOW 12817 HOURS
```

a. Structural Element: WSC-SWB-AFT-0000 (Concluded)

```
                              AIRCRAFT TYPE: HYBRID

      NUMBER OF AIRCRAFT IN FLEET:   500                    AIRCRAFT SERVICE LIFE:   60000 HOURS

                           STRUCTURAL ELEMENT: WSC-SWB-AFT-0030

PREDICTED AVERAGE FATIGUE LIFE: 690000 HOURS              ACTUAL AVERAGE FATIGUE LIFE:  424261 HOURS

                               INITIAL INSPECTION INTERVALS

                                  A-LEVEL    25 HOURS
                                  B-LEVEL   200 HOURS
                                  C-LEVEL  1000 HOURS
                                  D-LEVEL 12000 HOURS

A/C NO.  86 ENTERS SERVICE   4400 HOURS FROM START OF SIMULATION

    1ST CRACK INITIATION PROJECTED AT  514080 FLIGHT HOURS
    2ND CRACK INITIATION PROJECTED AT  644584 FLIGHT HOURS
    3RD CRACK INITIATION PROJECTED AT  691441 FLIGHT HOURS
    SLOW CRACK GROWTH RATE = .000073 INCHES/HOUR
    FAST CRACK GROWTH RATE = .001468 INCHES/HOUR

INSPECTION INTERVAL INCREASE IMPLEMENTED
    C-LEVEL INTERVAL NOW  1250 HOURS
    D-LEVEL INTERVAL NOW 15000 HOURS

D-LEVEL INSPECTION PERFORMED ON A/C NO.  86 AT  12000 HOURS

INSPECTION INTERVAL INCREASE IMPLEMENTED
    C-LEVEL INTERVAL NOW  1563 HOURS
    D-LEVEL INTERVAL NOW 18750 HOURS

D-LEVEL INSPECTION PERFORMED ON A/C NO.  86 AT  27000 HOURS

A/C NO. 497 ENTERS SERVICE  44750 HOURS FROM START OF SIMULATION

    1ST CRACK INITIATION PROJECTED AT  514414 FLIGHT HOURS
    2ND CRACK INITIATION PROJECTED AT  548384 FLIGHT HOURS
    3RD CRACK INITIATION PROJECTED AT  663849 FLIGHT HOURS
    SLOW CRACK GROWTH RATE = .000073 INCHES/HOUR
    FAST CRACK GROWTH RATE = .001468 INCHES/HOUR

INSPECTION INTERVAL INCREASE IMPLEMENTED
    C-LEVEL INTERVAL NOW  1953 HOURS
    D-LEVEL INTERVAL NOW 23438 HOURS

D-LEVEL INSPECTION PERFORMED ON A/C NO.  86 AT  45750 HOURS

D-LEVEL INSPECTION PERFORMED ON A/C NO. 497 AT  12000 HOURS

A/C NO.  86 RETIRED FROM SERVICE AT  60000 FLIGHT HOURS

D-LEVEL INSPECTION PERFORMED ON A/C NO. 497 AT  35438 HOURS

D-LEVEL INSPECTION PERFORMED ON A/C NO. 497 AT  58875 HOURS

A/C NO. 497 RETIRED FROM SERVICE AT  60000 FLIGHT HOURS
```

b. Structural Element: WSC-SWB-AFT-0030

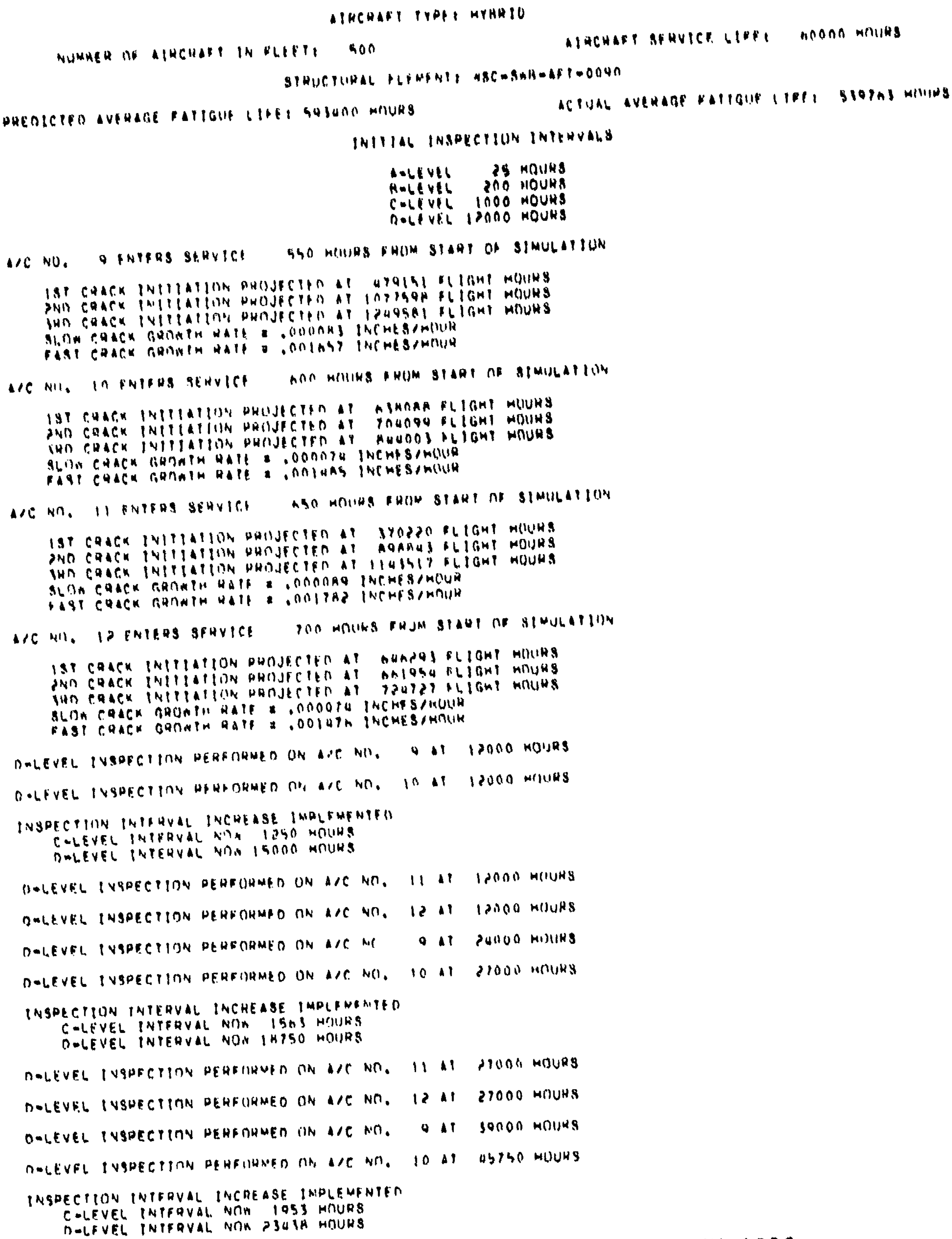
Figure 6 - Continued



```
                              AIRCRAFT TYPE: HYBRID

     NUMBER OF AIRCRAFT IN FLEET:   500               AIRCRAFT SERVICE LIFE:   60000 HOURS

                        STRUCTURAL ELEMENT: WSC-SWB-AFT-0060

PREDICTED AVERAGE FATIGUE LIFE: 662400 HOURS          ACTUAL AVERAGE FATIGUE LIFE:  547820 HOURS

                              INITIAL INSPECTION INTERVALS

                                   A-LEVEL     25 HOURS
                                   B-LEVEL    200 HOURS
                                   C-LEVEL   1000 HOURS
                                   D-LEVEL  12000 HOURS

A/C NO.    1 ENTERS SERVICE     150 HOURS FROM START OF SIMULATION

     1ST CRACK INITIATION PROJECTED AT  379222 FLIGHT HOURS
     2ND CRACK INITIATION PROJECTED AT 1099412 FLIGHT HOURS
     3RD CRACK INITIATION PROJECTED AT 1158883 FLIGHT HOURS
     SLOW CRACK GROWTH RATE = .000089 INCHES/HOUR
     FAST CRACK GROWTH RATE = .001778 INCHES/HOUR

D-LEVEL INSPECTION PERFORMED ON A/C NO.    1 AT  12000 HOURS

INSPECTION INTERVAL INCREASE IMPLEMENTED
     C-LEVEL INTERVAL NOW  1250 HOURS
     D-LEVEL INTERVAL NOW 15000 HOURS

D-LEVEL INSPECTION PERFORMED ON A/C NO.    1 AT  24000 HOURS

INSPECTION INTERVAL INCREASE IMPLEMENTED
     C-LEVEL INTERVAL NOW  1563 HOURS
     D-LEVEL INTERVAL NOW 18750 HOURS

D-LEVEL INSPECTION PERFORMED ON A/C NO.    1 AT  39000 HOURS

INSPECTION INTERVAL INCREASE IMPLEMENTED
     C-LEVEL INTERVAL NOW  1953 HOURS
     D-LEVEL INTERVAL NOW 23438 HOURS

INSPECTION INTERVAL DECREASE IMPLEMENTED
     C-LEVEL INTERVAL NOW   684 HOURS
     D-LEVEL INTERVAL NOW  8203 HOURS

FLEET WIDE SPECIAL INSPECTION PERFORMED

D-LEVEL INSPECTION PERFORMED ON A/C NO.    1 AT  53806 HOURS

INSPECTION INTERVAL INCREASE IMPLEMENTED
     C-LEVEL INTERVAL NOW   854 HOURS
     D-LEVEL INTERVAL NOW 10254 HOURS

A/C NO.    1 RETIRED FROM SERVICE AT  60000 FLIGHT HOURS

INSPECTION INTERVAL INCREASE IMPLEMENTED
     C-LEVEL INTERVAL NOW  1068 HOURS
     D-LEVEL INTERVAL NOW 12817 HOURS

INSPECTION INTERVAL INCREASE IMPLEMENTED
     C-LEVEL INTERVAL NOW  1335 HOURS
     D-LEVEL INTERVAL NOW 16022 HOURS

INSPECTION INTERVAL INCREASE IMPLEMENTED
     C-LEVEL INTERVAL NOW  1669 HOURS
     D-LEVEL INTERVAL NOW 20027 HOURS
```

c. Structural Element: WSC-SWB-AFT-0060

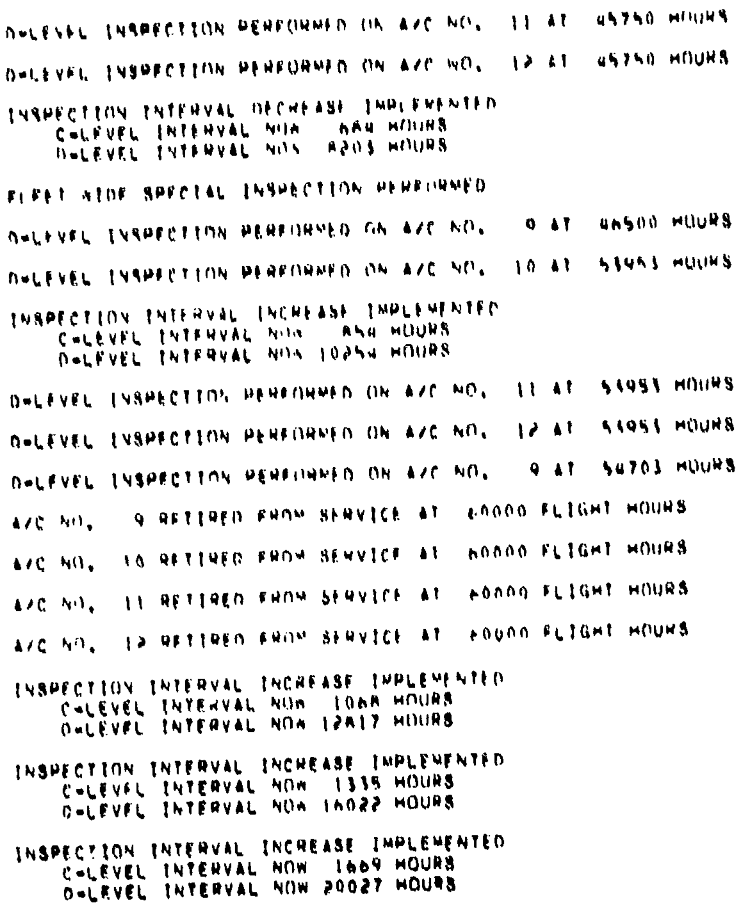
Figure 6 - Continued



```
                              AIRCRAFT TYPE: HYBRID

         NUMBER OF AIRCRAFT IN FLEET:   500                      AIRCRAFT SERVICE LIFE:   60000 HOURS

                               STRUCTURAL ELEMENT: WSC-SWB-AFT-0090

PREDICTED AVERAGE FATIGUE LIFE: 593400 HOURS                 ACTUAL AVERAGE FATIGUE LIFE:  519763 HOURS

                                   INITIAL INSPECTION INTERVALS

                                        A-LEVEL     25 HOURS
                                        B-LEVEL    200 HOURS
                                        C-LEVEL   1000 HOURS
                                        D-LEVEL  12000 HOURS

A/C NO.    9 ENTERS SERVICE      550 HOURS FROM START OF SIMULATION

     1ST CRACK INITIATION PROJECTED AT  479151 FLIGHT HOURS
     2ND CRACK INITIATION PROJECTED AT 1077598 FLIGHT HOURS
     3RD CRACK INITIATION PROJECTED AT 1249581 FLIGHT HOURS
     SLOW CRACK GROWTH RATE = .000083 INCHES/HOUR
     FAST CRACK GROWTH RATE = .001657 INCHES/HOUR

A/C NO.   10 ENTERS SERVICE      600 HOURS FROM START OF SIMULATION

     1ST CRACK INITIATION PROJECTED AT  638088 FLIGHT HOURS
     2ND CRACK INITIATION PROJECTED AT  704099 FLIGHT HOURS
     3RD CRACK INITIATION PROJECTED AT  844003 FLIGHT HOURS
     SLOW CRACK GROWTH RATE = .000074 INCHES/HOUR
     FAST CRACK GROWTH RATE = .001865 INCHES/HOUR

A/C NO.   11 ENTERS SERVICE      650 HOURS FROM START OF SIMULATION

     1ST CRACK INITIATION PROJECTED AT  370220 FLIGHT HOURS
     2ND CRACK INITIATION PROJECTED AT  898843 FLIGHT HOURS
     3RD CRACK INITIATION PROJECTED AT 1143517 FLIGHT HOURS
     SLOW CRACK GROWTH RATE = .000089 INCHES/HOUR
     FAST CRACK GROWTH RATE = .001782 INCHES/HOUR

A/C NO.   12 ENTERS SERVICE      700 HOURS FROM START OF SIMULATION

     1ST CRACK INITIATION PROJECTED AT  648293 FLIGHT HOURS
     2ND CRACK INITIATION PROJECTED AT  661954 FLIGHT HOURS
     3RD CRACK INITIATION PROJECTED AT  724727 FLIGHT HOURS
     SLOW CRACK GROWTH RATE = .000074 INCHES/HOUR
     FAST CRACK GROWTH RATE = .001476 INCHES/HOUR

D-LEVEL INSPECTION PERFORMED ON A/C NO.    9 AT   12000 HOURS

D-LEVEL INSPECTION PERFORMED ON A/C NO.   10 AT   12000 HOURS

INSPECTION INTERVAL INCREASE IMPLEMENTED
     C-LEVEL INTERVAL NOW  1250 HOURS
     D-LEVEL INTERVAL NOW 15000 HOURS

D-LEVEL INSPECTION PERFORMED ON A/C NO.   11 AT   12000 HOURS

D-LEVEL INSPECTION PERFORMED ON A/C NO.   12 AT   12000 HOURS

D-LEVEL INSPECTION PERFORMED ON A/C NC     9 AT   24000 HOURS

D-LEVEL INSPECTION PERFORMED ON A/C NO.   10 AT   21000 HOURS

INSPECTION INTERVAL INCREASE IMPLEMENTED
     C-LEVEL INTERVAL NOW  1563 HOURS
     D-LEVEL INTERVAL NOW 18750 HOURS

D-LEVEL INSPECTION PERFORMED ON A/C NO.   11 AT   21000 HOURS

D-LEVEL INSPECTION PERFORMED ON A/C NO.   12 AT   27000 HOURS

D-LEVEL INSPECTION PERFORMED ON A/C NO.    9 AT   39000 HOURS

D-LEVEL INSPECTION PERFORMED ON A/C NO.   10 AT   45750 HOURS

INSPECTION INTERVAL INCREASE IMPLEMENTED
     C-LEVEL INTERVAL NOW  1953 HOURS
     D-LEVEL INTERVAL NOW 23438 HOURS
```

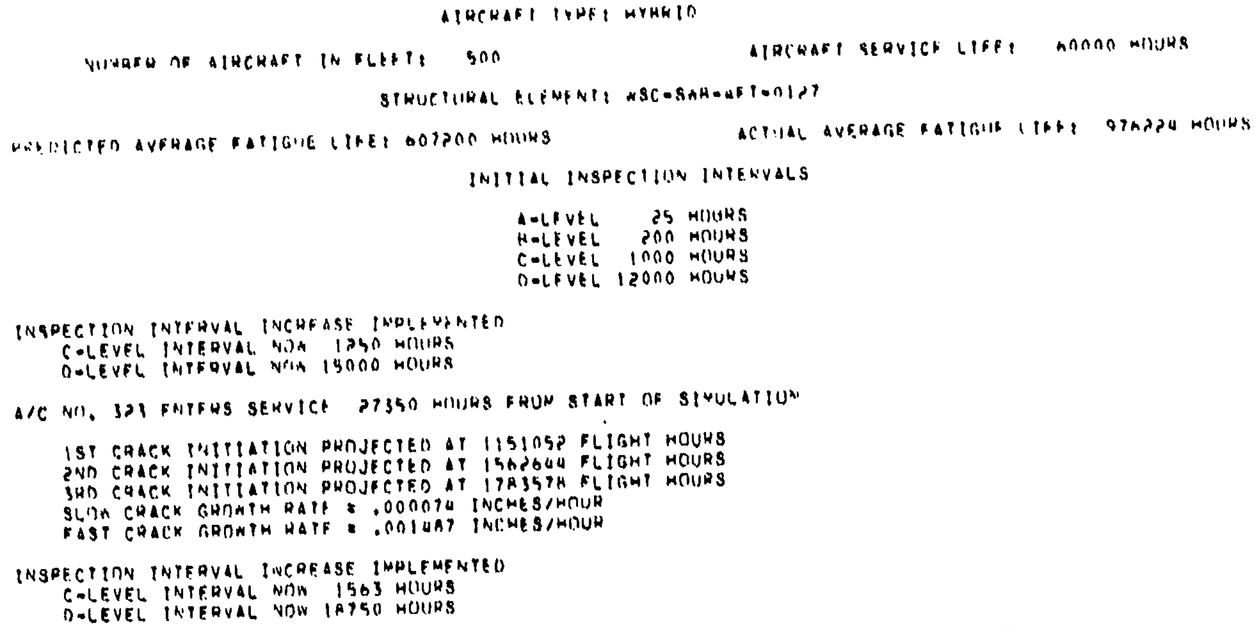
d. Structural Element: WSC-SWB-AFT-0090

Figure 6 - Continued



```
D-LEVEL INSPECTION PERFORMED ON A/C NO.  11 AT  45750 HOURS

D-LEVEL INSPECTION PERFORMED ON A/C NO.  12 AT  45750 HOURS

INSPECTION INTERVAL DECREASE IMPLEMENTED
    C-LEVEL INTERVAL NOW   684 HOURS
    D-LEVEL INTERVAL NOW  8203 HOURS

FLEET WIDE SPECIAL INSPECTION PERFORMED

D-LEVEL INSPECTION PERFORMED ON A/C NO.   9 AT  48500 HOURS

D-LEVEL INSPECTION PERFORMED ON A/C NO.  10 AT  53953 HOURS

INSPECTION INTERVAL INCREASE IMPLEMENTED
    C-LEVEL INTERVAL NOW   854 HOURS
    D-LEVEL INTERVAL NOW 10254 HOURS

D-LEVEL INSPECTION PERFORMED ON A/C NO.  11 AT  53953 HOURS

D-LEVEL INSPECTION PERFORMED ON A/C NO.  12 AT  53953 HOURS

D-LEVEL INSPECTION PERFORMED ON A/C NO.   9 AT  54703 HOURS

A/C NO.   9 RETIRED FROM SERVICE AT  60000 FLIGHT HOURS

A/C NO.  10 RETIRED FROM SERVICE AT  60000 FLIGHT HOURS

A/C NO.  11 RETIRED FROM SERVICE AT  60000 FLIGHT HOURS

A/C NO.  12 RETIRED FROM SERVICE AT  60000 FLIGHT HOURS

INSPECTION INTERVAL INCREASE IMPLEMENTED
    C-LEVEL INTERVAL NOW  1068 HOURS
    D-LEVEL INTERVAL NOW 12817 HOURS

INSPECTION INTERVAL INCREASE IMPLEMENTED
    C-LEVEL INTERVAL NOW  1335 HOURS
    D-LEVEL INTERVAL NOW 16022 HOURS

INSPECTION INTERVAL INCREASE IMPLEMENTED
    C-LEVEL INTERVAL NOW  1669 HOURS
    D-LEVEL INTERVAL NOW 20027 HOURS
```

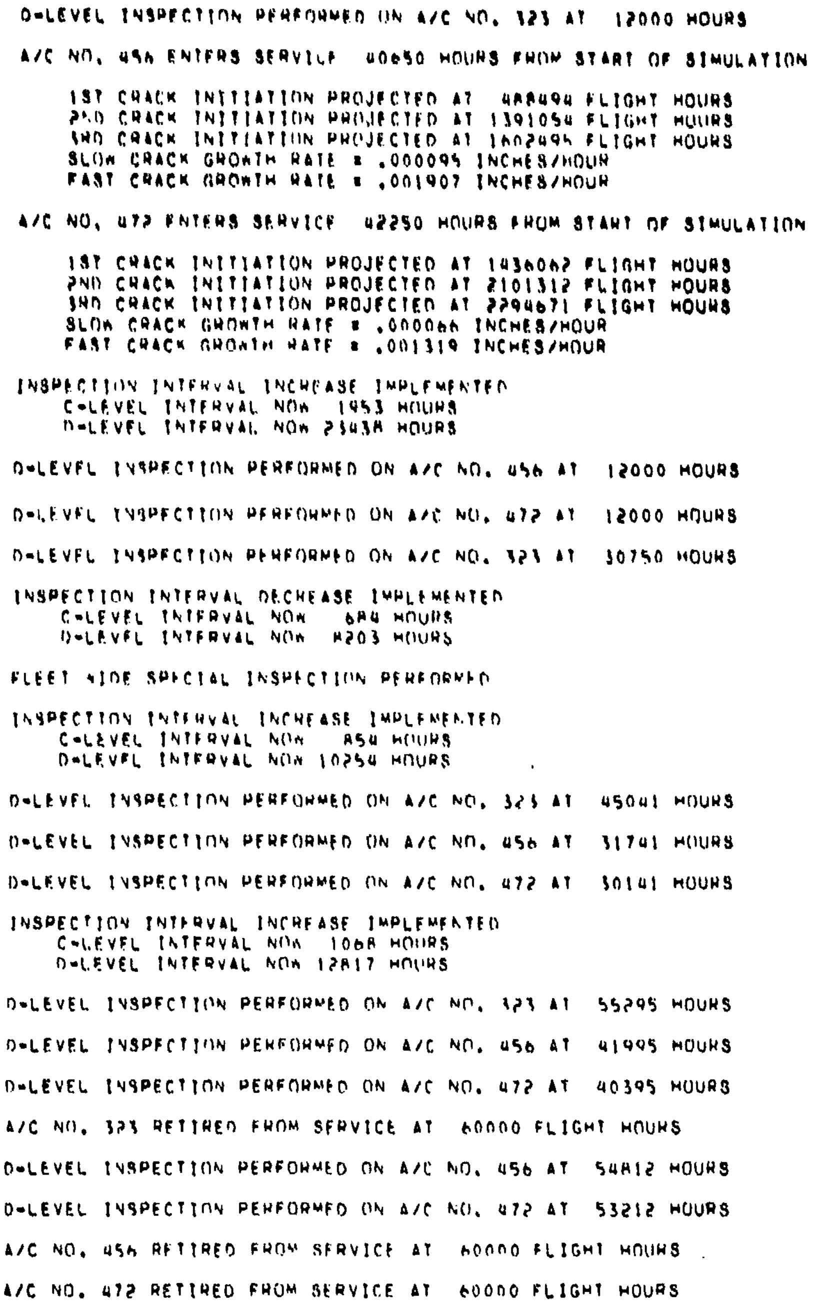
d. Structural Element: WSC-SWB-AFT-0090 (Concluded)



```
                              AIRCRAFT TYPE: HYBRID

      NUMBER OF AIRCRAFT IN FLEET:   500                 AIRCRAFT SERVICE LIFE:   60000 HOURS

                         STRUCTURAL ELEMENT: WSC-SWB-AFT-0127

PREDICTED AVERAGE FATIGUE LIFE: 607200 HOURS            ACTUAL AVERAGE FATIGUE LIFE:  976224 HOURS

                            INITIAL INSPECTION INTERVALS

                                A-LEVEL    25 HOURS
                                B-LEVEL   200 HOURS
                                C-LEVEL  1000 HOURS
                                D-LEVEL 12000 HOURS

INSPECTION INTERVAL INCREASE IMPLEMENTED
    C-LEVEL INTERVAL NOW  1250 HOURS
    D-LEVEL INTERVAL NOW 15000 HOURS

A/C NO. 323 ENTERS SERVICE  27350 HOURS FROM START OF SIMULATION

    1ST CRACK INITIATION PROJECTED AT 1151052 FLIGHT HOURS
    2ND CRACK INITIATION PROJECTED AT 1562644 FLIGHT HOURS
    3RD CRACK INITIATION PROJECTED AT 1783578 FLIGHT HOURS
    SLOW CRACK GROWTH RATE = .000074 INCHES/HOUR
    FAST CRACK GROWTH RATE = .001487 INCHES/HOUR

INSPECTION INTERVAL INCREASE IMPLEMENTED
    C-LEVEL INTERVAL NOW  1563 HOURS
    D-LEVEL INTERVAL NOW 18750 HOURS
```

e. Structural Element: WSC-SWB-AFT-0127

Figure 6 - Continued



```
D-LEVEL INSPECTION PERFORMED ON A/C NO. 323 AT  12000 HOURS

A/C NO. 456 ENTERS SERVICE  40650 HOURS FROM START OF SIMULATION

     1ST CRACK INITIATION PROJECTED AT  488494 FLIGHT HOURS
     2ND CRACK INITIATION PROJECTED AT 1391054 FLIGHT HOURS
     3RD CRACK INITIATION PROJECTED AT 1602495 FLIGHT HOURS
     SLOW CRACK GROWTH RATE = .000095 INCHES/HOUR
     FAST CRACK GROWTH RATE = .001907 INCHES/HOUR

A/C NO. 472 ENTERS SERVICE  42250 HOURS FROM START OF SIMULATION

     1ST CRACK INITIATION PROJECTED AT 1436062 FLIGHT HOURS
     2ND CRACK INITIATION PROJECTED AT 2101312 FLIGHT HOURS
     3RD CRACK INITIATION PROJECTED AT 2294671 FLIGHT HOURS
     SLOW CRACK GROWTH RATE = .000066 INCHES/HOUR
     FAST CRACK GROWTH RATE = .001319 INCHES/HOUR

INSPECTION INTERVAL INCREASE IMPLEMENTED
     C-LEVEL INTERVAL NOW  1953 HOURS
     D-LEVEL INTERVAL NOW 23438 HOURS

D-LEVEL INSPECTION PERFORMED ON A/C NO. 456 AT  12000 HOURS

D-LEVEL INSPECTION PERFORMED ON A/C NO. 472 AT  12000 HOURS

D-LEVEL INSPECTION PERFORMED ON A/C NO. 323 AT  30750 HOURS

INSPECTION INTERVAL DECREASE IMPLEMENTED
     C-LEVEL INTERVAL NOW   684 HOURS
     D-LEVEL INTERVAL NOW  8203 HOURS

FLEET WIDE SPECIAL INSPECTION PERFORMED

INSPECTION INTERVAL INCREASE IMPLEMENTED
     C-LEVEL INTERVAL NOW   854 HOURS
     D-LEVEL INTERVAL NOW 10254 HOURS

D-LEVEL INSPECTION PERFORMED ON A/C NO. 323 AT  45041 HOURS

D-LEVEL INSPECTION PERFORMED ON A/C NO. 456 AT  31741 HOURS

D-LEVEL INSPECTION PERFORMED ON A/C NO. 472 AT  30141 HOURS

INSPECTION INTERVAL INCREASE IMPLEMENTED
     C-LEVEL INTERVAL NOW  1068 HOURS
     D-LEVEL INTERVAL NOW 12817 HOURS

D-LEVEL INSPECTION PERFORMED ON A/C NO. 323 AT  55295 HOURS

D-LEVEL INSPECTION PERFORMED ON A/C NO. 456 AT  41995 HOURS

D-LEVEL INSPECTION PERFORMED ON A/C NO. 472 AT  40395 HOURS

A/C NO. 323 RETIRED FROM SERVICE AT  60000 FLIGHT HOURS

D-LEVEL INSPECTION PERFORMED ON A/C NO. 456 AT  54812 HOURS

D-LEVEL INSPECTION PERFORMED ON A/C NO. 472 AT  53212 HOURS

A/C NO. 456 RETIRED FROM SERVICE AT  60000 FLIGHT HOURS

A/C NO. 472 RETIRED FROM SERVICE AT  60000 FLIGHT HOURS
```

e. Structural Element: WSC-SWB-AFT-0127 (Concluded)

Figure 6 - Concluded

APPENDIX A

DETAILED PROGRAM DESCRIPTION

In the following detailed description of the SAIFE program, each event and routine in the program is presented separately. Except for the PREAMBLE, each presentation consists of a description, the definition of the local variables, if any, and a flow chart to illustrate the logic of the event or routine.

APPENDIX A

TABLE OF CONTENTS

1. PREAMBLE

The PREAMBLE is the definition section of a SIMSCRIPT program. All global variables and global arrays are defined. Temporary entities are defined and tally statistics are identified. Event notices and functions are defined and an event priority order is set. The global real variables are listed and described below in the order in which they appear in the PREAMBLE. Global real variables which are input variables are not included here but can be found in the input section.

Global Real Variables

FLEET.STR.RED - This variable is the sum of crack lengths found in the fleet since the last inspection frequency change.

1AAFL - This variable is the actual average fatigue life of the element design determined in the MAIN program.

CRRF - Assigned a value in routine INITIALIZATION according to the element corrosion resistance rating, this variable is multiplied by the aircraft corrosion growth rate to give the element corrosion growth rate.

COST.OF.REPAIRS - This variable is the sum of repair costs for the fleet since the last modification.

FIXIT.COST - This variable is the cost of repairing a defect found at a particular inspection level. Its value is set in the inspection events.

CHG.FREQ.TIME - This variable is set equal to TIME.V whenever an inspection interval change is scheduled in the event REPAIR.

1CRKT - Each time a first crack occurs, this variable is set equal to the service time on the aircraft.

1CORT - Each time corrosion occurs, this variable is set equal to the service time on the aircraft.

1SDT - Each time service damage occurs, this variable is set equal to the service time on the aircraft.

ACRKL, BCRKL, CCRKL, DCRKL, SCRKL - Each time a crack is found during an A-level, B-level, C-level, D-level, or Special inspection, the corresponding variable is set equal to the crack length.

ACA, BCA, CCA, DCA, SCA - Each time corrosion is found during an A-level, B-level, C-level, D-level, or Special inspection, the corresponding variable is set equal to the corrosion area.

AIRFRAME.TIME - This variable is the number of flight hours accumulated since the last modification for aircraft no longer in service.

G1CRK - Each time a first crack occurs, this variable is set equal to the service time on the aircraft.

G1COR - Each time corrosion occurs, this variable is set equal to the service time on the aircraft.

G1SD - Each time service damage occurs, this variable is set equal to the service time on the aircraft.

GACRK, GBCRK, GCCRK, GDCRK, GSCRK - Each time a crack is found during an A-level, B-level, C-level, D-level, or Special inspection, the corresponding variable is set equal to the crack length.

GACA, GBCA, GCCA, GDCA, GSCA - Each time corrosion is found during an A-level, B-level, C-level, D-level, or Special inspection, the corresponding variable is set equal to the corrosion area.

CINSL, DINSL - Each time there is an inspection interval change, these variables are set equal to the C-level and D-level intervals, respectively.

KSMP - This variable is set equal to 1.0 in the A-level, B-level, and C-level inspection events and set equal to the D-level sampling percentage in the D-level inspection event.

The global integer variables are listed next. Again, input variables are not included in this list.

Global Integer Variables

ID - In each event and routine, this variable is the identification number of the aircraft being processed.

IDCK - This variable is initialized to zero and incremented by one each time an aircraft enters service.

I - This variable is used as a local index or array subscript in different locations in the program.

COUNT.ELEMENT - Each time new element data is read in, this variable is incremented by one.

NICHG - This variable is the number of times that the inspection intervals have changed.

LHTA - This variable is the identification number of the aircraft among the ten high-time aircraft with the fewest flight hours.

1.NUM.OF.RETIRE - This variable is the number of aircraft that have been retired from service.

2.NUM.OF.CRASH - This variable is the number of aircraft which have been removed from service because of structural failure.

ITRNL - This variable is the numeric identification of the lowest internal level of inspection.

EXT.INSP.LEVEL - This variable is the numeric identification of the lowest external level of inspection.

IIL, ELL - These variables are the numeric identifications of the lowest internal and external levels of inspection, respectively. If either of these variables is less than three, it is set equal to three.

TO.BE.MODIFIED - This variable is the number of aircraft with a pending retrofit modification.

BEEN.MODIFIED - This variable is the number of aircraft that have had a current retrofit modification installed.

FDCK - This variable is the number of aircraft in service when a modification is implemented because of a fatigue test failure.

OFCR, OCOR, OSDM, OPD - These variables are the number of occurrences of first cracks, corrosion, service damage, and production defects, respectively, for a particular element.

OSCR, OSCO - These variables are the number of cracks and corrosion defects, respectively, detected during a special inspection for a particular element.

NSIC - This variable is the number of special inspections conducted for a particular element.

NSMD - This variable is the number of aircraft modified in service for a particular element.

NSFL - This variable is the number of aircraft experiencing structural failure for a particular element.

NMD - This variable is the number of structural modifications made on a particular element.

NRFS - This variable is the number of aircraft with the residual strength for a particular element reaching the fail-safe strength.

SNRES - This variable is the number of aircraft with the residual strength for a particular element type reaching the fail-safe strength.

J - This variable is used as a local index or array subscript in different locations in the program.

LDA - If the long list option is in effect, this variable is the ascending sequential position of the element being processed among those elements read in under the long list option.

GOICR, GOCOR, GOSDM - These variables are the number of occurrences of first cracks, corrosion damage, and service damage, respectively, for a particular element type.

GOSCR, GOSCO - These variables are the number of cracks and corrosion defects, respectively, detected during a special inspection for a particular element type.

GOPD - This variable is the number of occurrences of production defects for a particular element type.

SNSIC - This variable is the number of special inspections conducted for a particular element type.

SNMD - This variable is the number of structural modifications made on a particular element type.

SNSMD - This variable is the number of aircraft modified in service for a particular element type.

SNSFL - This variable is the number of aircraft experiencing structural failure for a particular element type.

The real arrays are listed next. Unless otherwise noted, all arrays are 1-dimensional. As before, input arrays are not included in this list.

Real Arrays

C.INTERVAL, D.INTERVAL - The elements of these arrays are the current C-level and D-level inspection intervals for each aircraft in the fleet.

ABCD - This array is of size four and contains the most recent intervals for each of the four levels of inspection.

CKREP.TIME - This array is the simulation time of the most recent crack repair for each aircraft.

CORRP.TIME - This array is the simulation time of the most recent corrosion repair for each aircraft.

LAST.SD - This array is the simulation time of the most recent occurrence of service damage for each aircraft.

OCCUR.MOD - This array is the simulation time when the most recent modification was installed for each aircraft.

MSR, MFR - These arrays are the slow and fast crack growth rates, respectively, for each aircraft.

SC, SD - These arrays contain each of the inspection interval changes for the C-level and D-level, respectively.

XRN - This array is the random number selected to calculate the time until structural failure for each aircraft.

MRDD - This array is the simulation time of the most recently detected defect at either a C-level or D-level inspection for each of the ten high-time aircraft.

GCRI - This array is the corrosion multiplying factor for each aircraft.

The following are the integer arrays. Again, unless otherwise noted, all arrays are 1-dimensional and input arrays are not included.

Integer Arrays

A1SR, A2SR, A3SR - These arrays contain the event notice identification numbers for each aircraft for the events 1.STRENGTH. REDUCTION, 2.STRENGTH.REDUCTION, and 3.STRENGTH.REDUCTION, respectively.

AF - This array is the event notice identification number for each aircraft for event FAILURE.

AIRPLANE - This array is the temporary entity identification number for each aircraft.

AAL, ABL, ACL, ADL - These arrays are the event notice identification numbers for each aircraft for the events A.LEVEL.INSPECTION, B.LEVEL.INSPECTION, C.LEVEL.INSPECTION, and D.LEVEL. INSPECTION, respectively.

AC, ATII - These arrays are the event notice identification numbers for each aircraft for events COROSION and T.INSPECTION. INCREASE, respectively.

ACID - This array contains the identification numbers of those aircraft experiencing structural failure for a particular element.

OICR, OICO - These arrays are the number of cracks and corrosion defects, respectively, detected at each of the four levels of inspection for a particular element.

SACID - This array contains the identification numbers of those aircraft experiencing structural failure for a particular element type.

GOICR, GOICO - These arrays are the number of cracks and corrosion defects, respectively, detected at each of the four levels of inspection for a particular element type.

HI.TIME.ACRFT - This array contains the identification numbers of the ten high-time aircraft.

APID - This array contains the identification numbers of those aircraft with a particular element whose residual strength has reached the fail-safe strength.

SAPID - This array contains the identification numbers of those aircraft with a particular element type whose residual strength has reached the fail-safe strength.

STIM - This array contains the flight hours on each aircraft when the residual strength for a particular element reaches the fail-safe strength.

SSTIM - This array contains the flight hours on each aircraft when the residual strength for a particular element type reaches the fail-safe strength.

FLTHR - This array contains the flight hours on each aircraft when structural failure occurs for a particular element.

SFLTHR - This array contains the flight hours on each aircraft when structural failure occurs for a particular element type.

ARFSL - This array is the event notice identification number for each aircraft for event REACH.FAIL.SAFE.LGT.

A1E, A2E, A3E - These arrays are the event notice identification numbers for each aircraft for events 1.ITE, 2.ITE, and 3.ITE, respectively.

The global alpha arrays are listed next. As before, input arrays are not included in this list.

Global Alpha Arrays

1.CR.EXISTS, 2.CR.EXISTS, 3.CR.EXISTS - The elements of these arrays are set equal to "YES" for each aircraft whenever there is a first, second, and third crack initiation, respectively.

CO.EXISTS - This array is set equal to "YES" for each aircraft when it has corrosion initiation.

SD.SCH - This array is set equal to "YES" for each aircraft that has event IN.SERVICE.DAMAGE scheduled.

SSTAN - This array is the station number which identifies each aircraft experiencing the failure of a particular element type.

SELNB - This array is the station number which identifies each aircraft with a particular element type whose residual strength has reached the fail-safe strength.

AIL, FSH - This array is set equal to "YES" for each aircraft when events REACH.FAIL.SAFE.LGT and FAILURE, respectively, are scheduled.

IE1, IE2, IE3 - This array is set equal to "YES" for each aircraft that has events 1.ITE, 2.ITE, and 3.ITE, respectively, scheduled.

TMOD.PENDING - This array is set equal to "YES" for each aircraft that has a modification pending because of a fatigue test failure.

SMOD.PENDING - This array is set equal to "YES" for each aircraft that has a modification pending because of service experience.

INSP.SCH - This array is set equal to "YES" for each aircraft that has inspections below the overhaul level scheduled.

1.INT, 2.INT, 3.INT - These arrays are set equal to "YES" for each aircraft that has a first crack, second crack, or third crack, respectively, initiated internally.

C.INT - This array is set equal to "YES" for each aircraft that has corrosion initiated internally.

The temporary entity definitions and tally statements are self-explanatory. The events, functions, and routines are described in detail in the following sections.

2. MAIN

2.1 Description

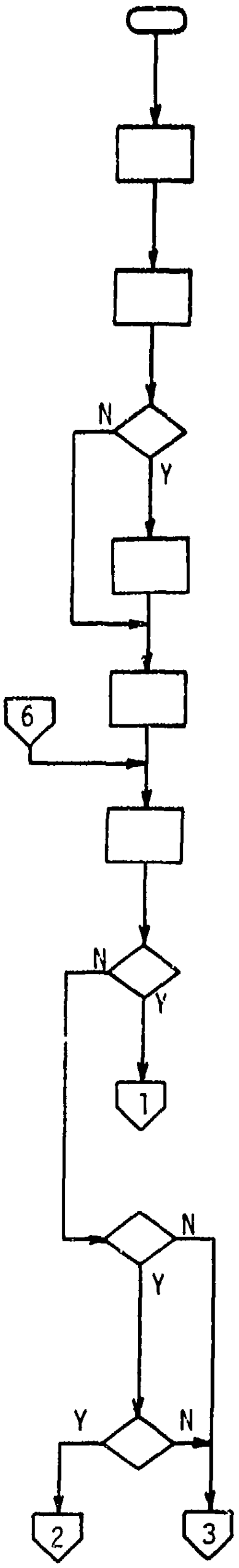
In the MAIN program, space is reserved for all global arrays. The following operations are performed in the order given: all input data is read in; the actual average fatigue life of the element type is calculated; the necessity of a structural modification because of a fatigue test failure is determined; the first event ENTER.SERVICE is scheduled; and the simulation is initiated.

2.2 Local Variables

NFTS - This real variable is the time in flight hours from when the second production rate goes into effect to when the last aircraft enters service.

SATL - This real variable is the earliest simulation time at which a structural modification because of a fatigue test failure is ready for installation.

2.3 Flow Chart



MAIN.

Reserve aircraft arrays.

Read aircraft data.

Does LONG.LIST="YES" ?

Read in elements and aircraft to be tracked.

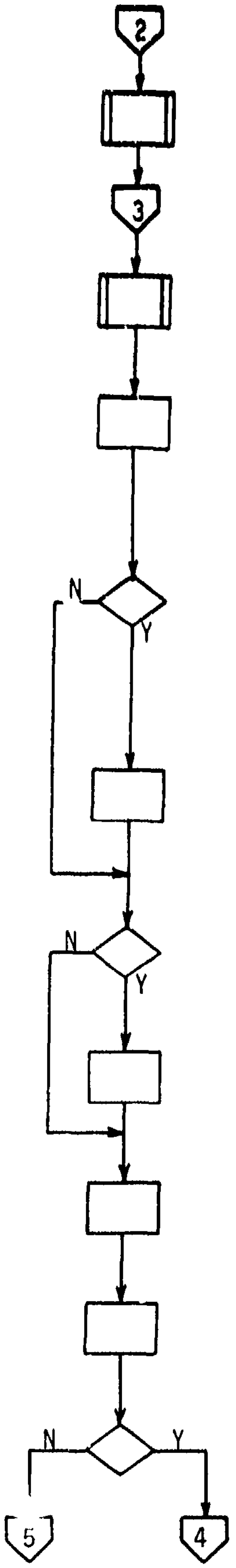
Reserve element arrays.

Read element data.

Does ELEMENT(1)="EOD" ?

Is this element a different element type than previous element?

Does LONG.LIST="NO"?



Call routine SUMMARY.

Call routine INITIALIZATION.

Calculate actual average fatigue life of element type.

Does fatigue test failure occur at less than twice the aircraft service life?

Schedule a structural modification.

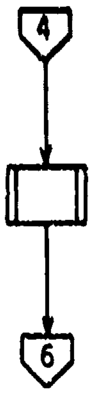
Is long list option in effect?

Print long list headings.

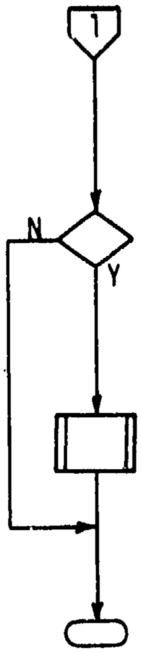
Schedule event ENTER.SERVICE.

Start simulation.

Does LONG.LIST="NO"?



Call routine DISPLAY.OUTPUT.



Does LONG.LIST="NO" ?

Call routine SUMMARY.

END.

3. INITIALIZATION

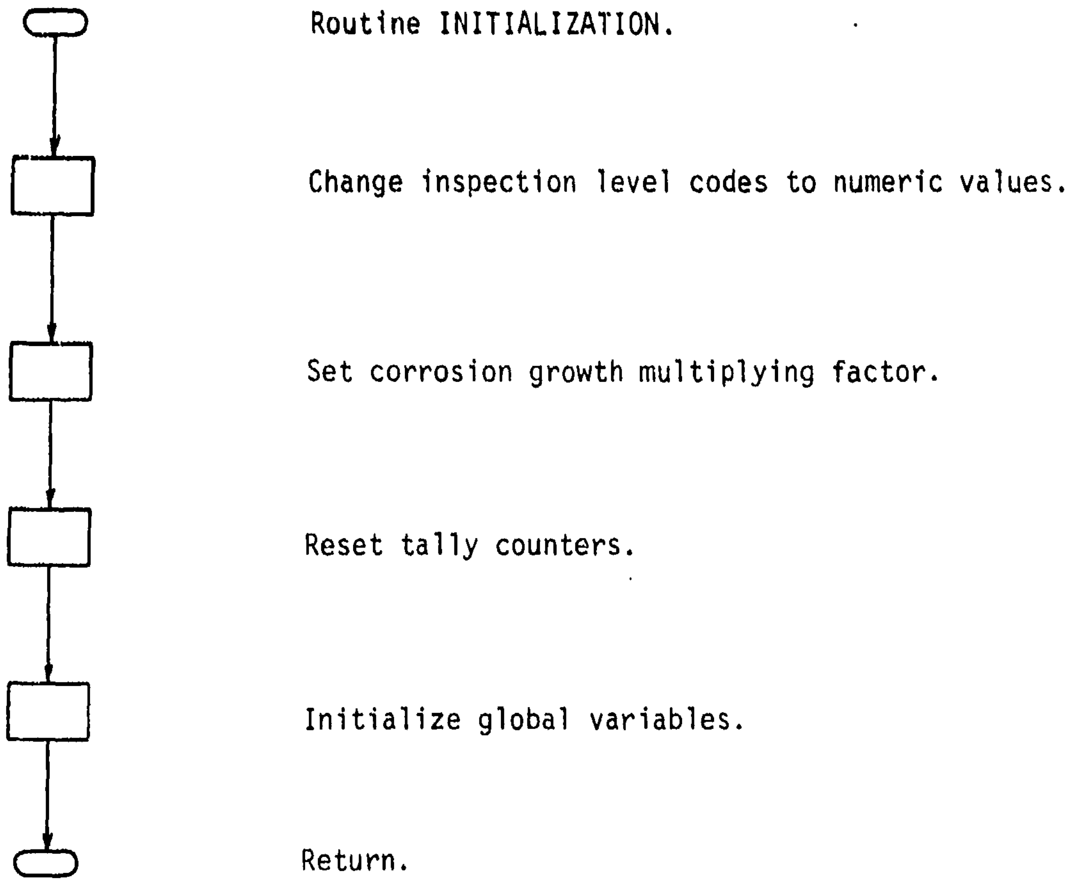
3.1 Description

This routine is called immediately after reading each new set of element input data. This routine changes the inspection level codes to numeric values, sets the corrosion growth multiplying factor based on the corrosion resistance rating, and resets the tally counters. It also initializes all the element global variables which are not part of the input. This routine is called from the MAIN program.

3.2 Local Variables

There are no local variables in this routine.

3.3 Flow Chart


Routine INITIALIZATION.
Change inspection level codes to numeric values.
Set corrosion growth multiplying factor.
Reset tally counters.
Initialize global variables.
Return.

4. SUM.INITIALIZE

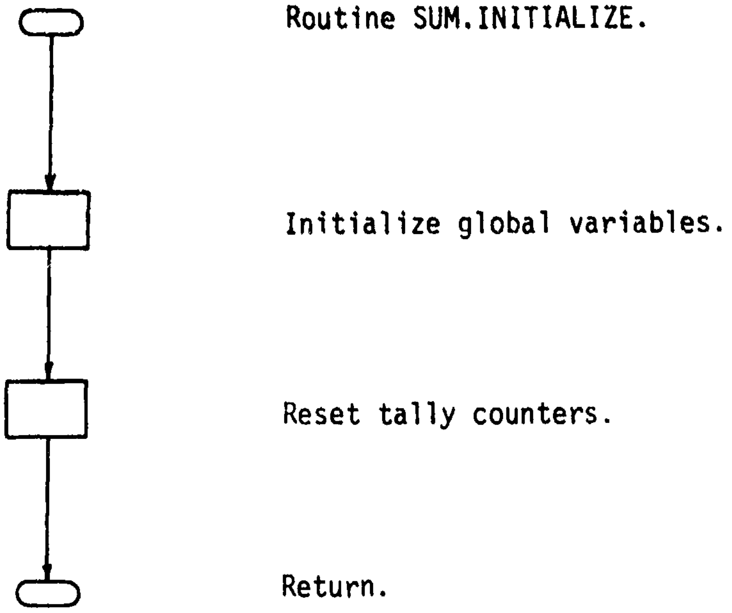
4.1 Description

This routine is called each time a new element type is read in. The element type is identified by the first twelve characters of the element identification. This routine initializes the global variables and resets the tally counters. This routine is called from the MAIN program.

4.2 Local Variables

There are no local variables in this routine.

4.3 Flow Chart



5. REAL.LIFE

5.1 Description

This routine accepts (1) the predicted average fatigue life of a particular element design and (2) the mean and standard deviation of the log-normal distribution of the ratio of the actual average fatigue life to the predicted average fatigue life. A random selection is made from the distribution and multiplied by the predicted average fatigue life. The resulting actual average fatigue life is returned to the calling routine. REAL.LIFE can be called from the MAIN program and events IMPLEMENT.MODIFICATION and T.IMPLEMENT.MOD.

5.2 Local Variables

MEAN - This real variable, whose value is passed from the calling routine, is the mean of the ratio distribution.

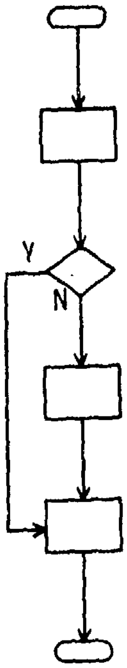
RATIO - This real variable, determined to be log-normally distributed, is the ratio of the actual fatigue life of an element design to its predicted fatigue life.

STD.DEV - This real variable, whose value is passed from the calling routine, is the standard deviation of the ratio distribution.

PDL - This real variable is the design predicted average fatigue life passed from the calling routine.

RFL - This real variable is the element actual average fatigue life which is returned to the calling routine.

5.3 Flow Chart



Routine REAL.LIFE.

Randomly select RATIO from distribution.

Is RATIO within allowable limits?

Set RATIO to limit.

Calculate actual average fatigue life.

Return with actual average fatigue life.

6. ENTER.SERVICE

6.1 Description

This event represents the entry into service of a new aircraft. The temporary entity AIRCRAFT is created and identified by the variable AIRPLANE(ID). The entity attributes TAIL.ID and ENTRY.TIME are defined and the AIRPLANE is filed in the set ACTIVE.FLEET. The routine FATIGUE.LIFE.SCATTER is called to determine the times to first, second, and third crack initiations. The slow and fast crack growth rates are calculated. The times to corrosion initiation and service damage are calculated. If either of these times is less than the service life of the aircraft, the corresponding defect is scheduled. If there is a production defect, the time to first crack initiation is replaced by a time drawn from a distribution of times to crack initiation of aircraft with production defects. If the long list option is in effect for each aircraft being tracked, this routine prints the following: (1) aircraft identification and time it enters service, (2) times to crack initiations, and (3) slow and fast crack growth rates. Crack initiations, D-level inspection, and retirement from service are also scheduled. If the present aircraft is not the last aircraft of the fleet, another ENTER. SERVICE is scheduled. This event can only be scheduled in the MAIN program and within itself.

6.2 Local Variables

DEFECT.LIFE - This real variable is the time to first crack initiation when the aircraft has a production defect.

HOURS.TO.CORROSION - This real variable is the time to corrosion initiation.

SECOND.LIFE - This real variable is the time to second crack initiation.

STD.SLOW - This real variable is the standard deviation of the distribution of slow crack growth rates.

FIRST.LIFE - This real variable is the time to first crack initiation when the aircraft has no production defect.

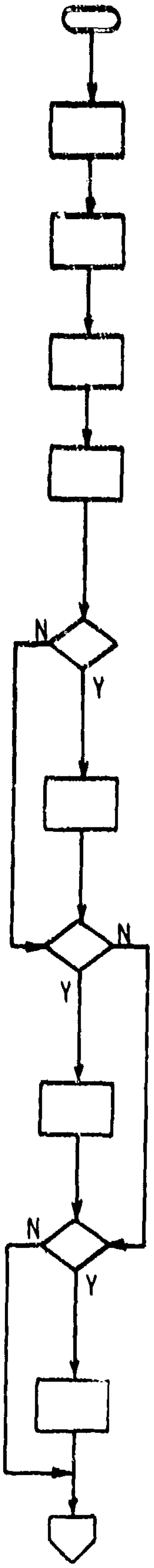
OURS.TO.SERVICE.DAMAGE - This real variable is the time to service damage occurrence.

RN - This real variable is a uniformly distributed random number between zero and one.

STD.FAST - This real variable is the standard deviation of the distribution of fast crack growth rates.

THIRD.LIFE - This real variable is the time to third crack initiation.

6.3 Flow Chart



Event ENTER.SERVICE.

Create an AIRCRAFT and file in ACTIVE.FLEET.

Determine times to first three crack initiations.

Calculate slow and fast crack growth rates.

Determine times to corrosion initiation and service damage.

Is time to corrosion initiation less than aircraft service life?

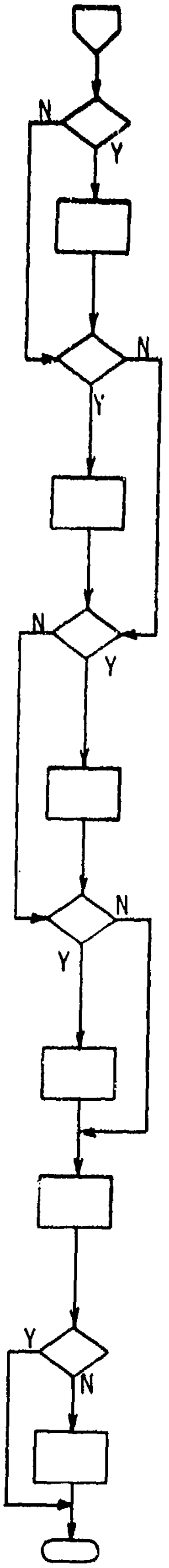
Schedule a corrosion initiation.

Is time to service damage less than aircraft service life?

Schedule a service damage occurrence.

Is random number less than probability of a production defect?

Determine new time to first crack initiation.



Is long list option in effect?

Print long list output if aircraft is one being tracked.

Is time to first crack initiation less than aircraft service life or is corrosion initiation scheduled?

Schedule first crack initiation.

Is time to second crack initiation less than aircraft service life or is corrosion initiation scheduled?

Schedule second crack initiation.

Is time to third crack initiation less than aircraft service life or is corrosion initiation scheduled?

Schedule third crack initiation.

Schedule first D-level inspection and aircraft retirement.

Is this last aircraft to enter service?

Schedule event ENTER.SERVICE.

Return.

7. FATIGUE.LIFE.SCATTER

7.1 Description

This routine receives the actual average fatigue life of the element design from the calling routine and returns the times to crack initiation of the first three fatigue cracks for the element in a particular aircraft. These times are random selections from a two-parameter Weibull distribution. This routine can be called from routine INSTALL.MODIFICATION and events ENTER.SERVICE and REPAIR.

7.2 Local Variables

ALPHA - This real variable is the shape parameter of the fatigue life distribution.

FIRST.LIFE - This real variable is the time to first crack initiation. This time is returned to the calling routine.

N - This integer variable, passed from the calling routine, identifies the random number stream to be used.

RN - This real variable is a uniformly distributed random number.

THIRD.LIFE - This real variable is the time to third crack initiation. This time is returned to the calling routine.

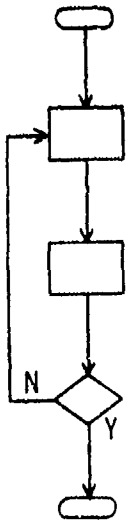
BETA - This real variable is the scale parameter of the fatigue life distribution.

LIFE - This real array of length three is used to temporarily hold the times to crack initiation of the three fatigue cracks.

RFL - This real variable is the element actual average fatigue life passed from the calling routine.

SECOND.LIFE - This real variable is the time to second crack initiation. This time is returned to the calling routine.

7.3 Flow Chart



Routine FATIGUE.LIFE.SCATTER.

Draw uniformly distributed random number.

Calculate time to crack initiation.

Times calculated for [illegible] cracks?

Return.

8. INSTALL.MODIFICATION

8.1 Description

This routine represents the installation of a structural modification caused by a fatigue test failure or by aircraft service experience. The modification is installed during a repair or a D-level inspection. All previously scheduled defect initiations are cancelled, and new times to defect initiations are calculated for each aircraft when it is modified. This routine can be called from the events REPAIR and D.LEVEL.INSPECTION.

8.2 Local Variables

DEFECT.LIFE - This real variable is the time to crack initiation drawn from a distribution of fatigue lives of elements having production defects.

HOURS.TO.CORROSION - This real variable contains the value returned by routine PREDICT.CORROSION.

RST - This real variable is the remaining service time to retirement of the aircraft being considered.

STD.FAST - This real variable is the standard deviation of the distribution of fast crack growth rates.

THIRD.LIFE - This real variable is the time to third crack initiation returned by routine FATIGUE.LIFE.SCATTER.

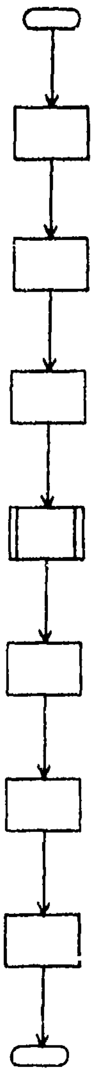
FIRST.LIFE - This real variable is the time to first crack initiation returned by routine FATIGUE.LIFE.SCATTER.

RN - This real variable is drawn from a uniform distribution of random numbers between 0 and 1.

SECOND.LIFE - This real variable is the time to second crack initiation returned by routine FATIGUE.LIFE.SCATTER.

STD.SLOW - This real variable is the standard deviation of the distribution of slow crack growth rates.

8.3 Flow Chart



Routine INSTALL.MODIFICATION.

Set C-level and D-level intervals to initial values.

Cancel previously scheduled inspections.

Cancel previously scheduled FAILURE.

Call routine FATIGUE.LIFE.SCATTER.

Cancel previously scheduled defect initiations.

Cancel previously scheduled REACH.FAIL.SAFE.LGT.

Schedule new defect initiations.

Return.

9. IN.SERVICE.DAMAGE

9.1 Description

This event represents the occurrence of a service damage defect. This occurrence results in the immediate initiation of the next scheduled crack. A new time to service damage is determined. If the new time is less than the remaining time in service of the aircraft, this event is scheduled once again. This event can be scheduled from within itself or in event ENTER.SERVICE.

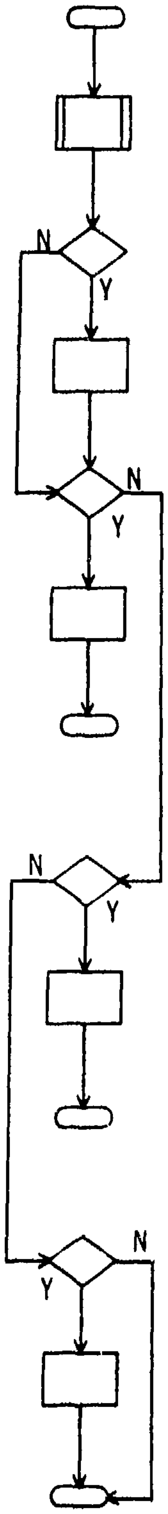
9.2 Local Variables

IDSDM - This integer variable is the identification number of the aircraft for which the event was scheduled.

RST - This real variable is the remaining service time to retirement of the aircraft being considered.

OURS.TO.SERVICE.DAMAGE - This real variable is the value returned by routine PREEDICT.SERVICE.DAMAGE.

9.3 Flow Chart



Event IN.SERVICE.DAMAGE.

Call routine PREEDICT.SERVICE.DAMAGE.

Is time to service damage less than remaining service time of aircraft?

Schedule event IN.SERVICE.DAMAGE.

Is first crack initiation scheduled?

Reschedule first crack to occur immediately.

Return.

Is second crack initiation scheduled?

Reschedule second crack to occur immediately.

Return.

Is third crack initiation scheduled?

Reschedule third crack to occur immediately.

Return.

10. T.IMPLEMENT.MOD

10.1 Description

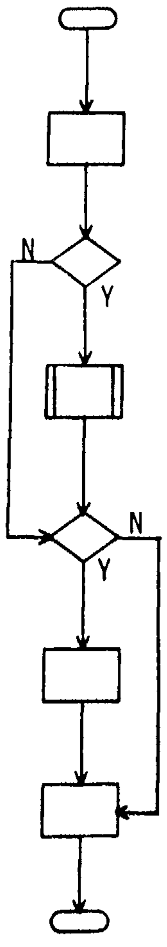
This event represents the development of a structural modification because of a fatigue test failure. This event determines the actual average fatigue life of the modification and schedules an increase in inspection frequencies at some percentage of the fatigue test life. This event is scheduled in the MAIN program.

10.2 Local Variables

NSIG - This real variable is (SIG.R)(.85) and is based on the assumption that a modification usually improves the actual average fatigue life of a particular design.

NMU - This real variable is MU.R+0.15(1.0-MU.R) and is also based on the foregoing assumption for NSIG.

10.3 Flow Chart



Event T.IMPLEMENT.MOD.

Set actual average fatigue life equal to predicted.

Does MOD.TESTED="NO"?

Call routine REAL.LIFE.

Is fatigue test life less than aircraft service life?

Indicate that aircraft in service require retrofit.

Schedule event T.INSPECTION.INCREASE.

Return.

11. PREEDICT.SERVICE.DAMAGE

11.1 Description

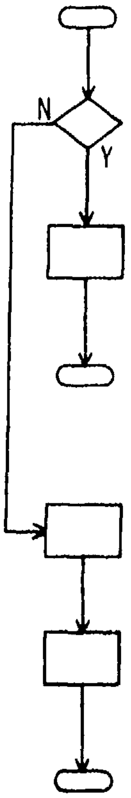
This routine generates the time to service damage occurrence for a given aircraft from a constant service damage occurrence rate. If the service damage occurrence rate is zero in the input, the routine sets the time to service damage occurrence as twice that of the aircraft service life. This routine can be called from events ENTER.SERVICE and IN.SERVICE.DAMAGE.

11.2 Local Variables

OURS.TO.SERVICE.DAMAGE - This real variable is the time to service damage occurrence in flight hours. This time is returned by the routine.

RN - This real variable is a uniformly distributed random number between 0 and 1.

11.3 Flow Chart



Routine PREEDICT.SERVICE.DAMAGE.

Is service damage occurrence rate $\leq$ 0?

Set time to service damage equal to twice aircraft service life.

Return.

Select uniformly distributed random number.

Calculate time to service damage.

Return.

12. PREDICT.CORROSION

12.1 Description

This routine generates time to corrosion initiation for a given aircraft from a time-dependent occurrence rate approximated by two constant rates. The first constant occurrence rate, the second constant occurrence rate, and the service time on the aircraft when the second rate goes into effect are all input variables. This routine can be called from the routine INSTALL. MODIFICATION and events ENTER.SERVICE and REPAIR.

12.2 Local Variables

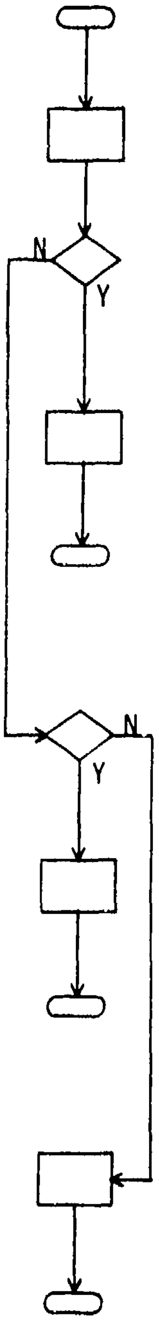
CRCT - This real variable is the remaining time in flight hours until the second corrosion occurrence rate goes into effect. This variable can be negative indicating that the second rate is already in effect.

LD - This real variable is used to hold an intermediate value during the calculation of time to corrosion initiation. The calculation uses a combination of both corrosion occurrence rates.

HOURS.TO.CORROSION - This real variable is the flight time until corrosion initiation. This time is returned to the calling routine.

RN - This real available is a uniformly distributed random number.

12.3 Flow Chart



Routine PREDICT.CORROSION.

Select uniformly distributed random number.

Is time on aircraft greater than time at which second occurrence rate goes into effect?

Calculate time to corrosion initiation using second rate.

Return.

Is selected random number > random number required to yield corrosion initiation at time of occurrence rate change?

Calculate time to corrosion initiation using first rate.

Return.

Calculate time to corrosion initiation using combination of both occurrence rates.

Return.

13. COROSION

13.1 Description

This event represents the initiation of a corrosion defect. The remaining time to crack initiation of all scheduled cracks is reduced by a corrosion damage factor. If either of the events FAILURE or REACH.FAIL.SAFE.LGT is scheduled, its remaining time until occurrence is also reduced by the corrosion damage factor. This event can be scheduled in the routine INSTALL. MODIFICATION and in events ENTER.SERVICE and REPAIR.

13.2 Local Variables

CDM.MULTIPLYING.FACTOR - This real variable is the factor which when multiplied by the remaining time to crack initiation accounts for the shortening effect of corrosion on fatigue lives.

NFTM - If a FAILURE has been scheduled, this real variable is the remaining time until its occurrence.

REMAINING.LIFE - This real variable is the remaining time until a scheduled crack initiation.

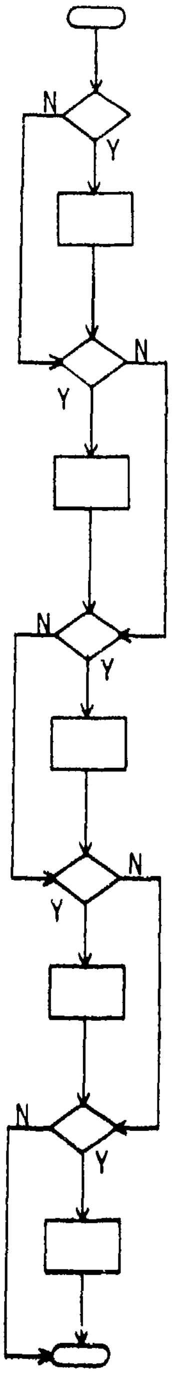
TRT - If a REACH.FAIL.SAFE.LGT has been scheduled, this real variable is the remaining time until its occurrence multiplied by the corrosion damage factor.

IDCO - This integer variable contains the identification number of the aircraft for which the event CORROSION was scheduled.

REDUCED.REMAINING.LIFE - This real variable is the REMAINING. LIFE multiplied by the corrosion damage factor.

RST - This real variable is the remaining service time of the aircraft under consideration.

13.3 Flow Chart



Event COROSION .

Is event REACH.FAIL.SAFE.LGT scheduled?

Reduce remaining time to occurrence of REACH.FAIL.SAFE.LGT by corrosion damage factor.

Is event FAILURE scheduled?

Reduce remaining time to occurrence of FAILURE by corrosion damage factor.

Is first crack initiation scheduled?

Reduce remaining time to first crack initiation by corrosion damage factor.

Is second crack initiation scheduled?

Reduce remaining time to second crack initiation by corrosion damage factor.

Is third crack initiation scheduled?

Reduce remaining time to third crack initiation by corrosion damage factor.

Return.

14. RATE

14.1 Description

This routine statistically generates element crack growth rates which reflect variation in material properties and load environment. The growth rates are randomly drawn from a normal distribution which is defined by a mean growth rate and a standard deviation passed from the calling routine. If a random draw yields a negative growth rate, the rate is set equal to the mean growth rate minus four standard deviations. Thus, the user must be sure that the standard deviation is always less than one-fourth of the mean. This routine is defined as a function in the routine PREAMBLE and is used in event ENTER.SERVICE and routine INSTALL.MODIFICATION.

14.2 Local Variables

G1 - This real variable is used to hold intermediate values in the calculation of the crack growth rate.

M - This real variable is the mean crack growth rate passed from the calling routine.

RN - This real variable is a uniformly distributed random number between 0 and 1.

S - This real variable is the crack growth rate standard deviation passed from the calling routine.

Z - This real variable is the element crack growth rate returned to the calling routine.

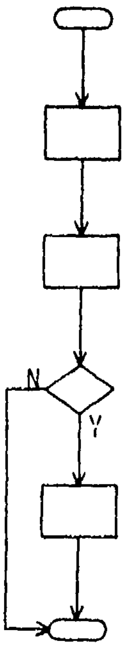
G2 - This real variable is used to hold intermediate values in the calculation of the crack growth rate.

N - This integer variable, passed from the calling routine, identifies the random number stream to be used.

RNI - This real variable, equal to 1.0 - RN, is a uniformly distributed random number between 0 and 1.

W - This real variable is used to hold intermediate values in the calculation of the crack growth rate.

14.3 Flow Chart



Routine RATE.

Select uniformly distributed random number.

Calculate crack growth rate.

Is calculated growth rate negative?

Set growth rate equal to mean minus four standard deviations.

Return.

15. 1.STRENGTH.REDUCTION

15.1 Description

This event represents the initiation of the first crack. If the long list option is in effect, the aircraft identification number and flight hours are printed for those aircraft being tracked. A uniform random number is compared with the probability of internal cracking to determine whether this crack initiates internally. If it does initiate internally, the time until it becomes external is calculated and the event 1.ITE is scheduled. Next, the time to structural failure is calculated by using the three-part residual strength curve described in Vol. II. If this time is less than the remaining service life of the aircraft, the event FAILURE is scheduled. The time until the residual strength of the element reaches the fail-safe strength is calculated, and event REACH.FAIL.SAFE.LGT is scheduled. This event can be scheduled in events ENTER.SERVICE, INSTALL.MODIFICATION, and REPAIR.

15.2 Local Variables

ARG - This real variable is used as an intermediate value in the calculation of time until structural failure.

GR2 - This real variable is the fast crack growth rate.

ID1SR - This integer variable is the aircraft identification number.

K1, K11, K13, K4, K9, LG - These real variables are used as intermediate values in the calculation of time until structural failure.

LIST - This real variable is set equal to 1.0 when the long list option is in effect and the aircraft being processed is one of those being tracked.

R2 - This real variable is the strength degradation rate when the crack length is between the critical crack length and the fail-safe length.

SF - This real variable is the element fail-safe strength.

S1 - This real variable is the element residual strength when the crack is at the critical crack length.

TAR - This real variable is the simulation time at which the aircraft being processed retires from service.

T1 - This real variable is the time in flight hours for the crack to grow from its initiation to the critical crack length.

GR1 - This real variable is the element slow crack growth rate.

K10, K12, K2, K8, LGK5 - These real variables are used as intermediate values in the calculation of time until structural failure.

R1 - This real variable is the strength degradation rate from crack initiation to critical crack length.

R3 - This real variable is the strength degradation rate from fail-safe strength to structural failure.

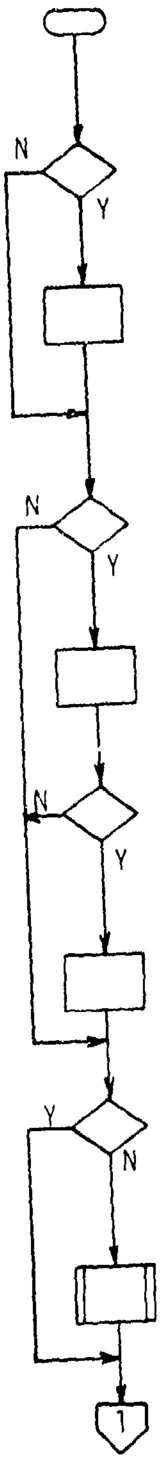
SU - This real variable is the ultimate strength of the element.

T - This real variable is the time in flight hours until a crack initiated internally becomes external.

TTF - This real variable is the time in flight hours until element failure.

T2 - This real variable is the time in flight hours until the fail-safe strength is reached.

15.3 Flow Chart



Event 1.STRENGTH.REDUCTION.

Is long list option in effect?

Print ID number and flight hours for aircraft being tracked.

Does crack initiate internally?

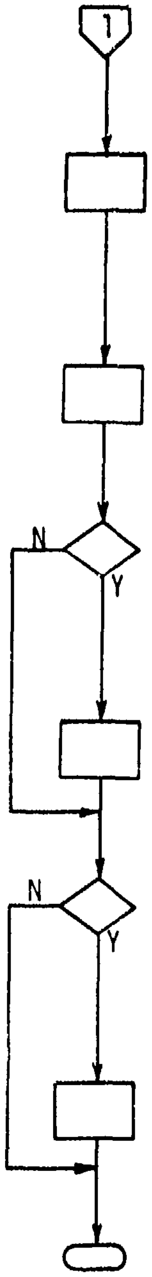
Set 1.INT to "YES".

Is time until crack becomes external less than remaining service life of aircraft?

Schedule event 1.ITE.

Is aircraft presently inspected below overhaul level?

Call routine INSPECTION.SCHEDULER.



Calculate time until residual strength equals fail-safe strength.

Calculate time until structural failure.

Is time until fail-safe strength less than remaining service life of aircraft?

Schedule event REACH.FAIL.SAFE.LGT.

Is time until failure less than remaining service life of aircraft?

Schedule event FAILURE.

Return.

16. 2.STRENGTH.REDUCTION

16.1 Description

This event represents the second crack initiation. If the long list option is in effect, the aircraft identification number and flight hours are printed for those aircraft being tracked. A uniform random number is compared with the probability of internal cracking to determine whether this crack initiates internally. If it does initiate internally, the time until it becomes external is calculated and the event 2.ITE is scheduled. Next, the time to structural failure is calculated by using the same three-part residual strength curve as in event 1.STRENGTH.REDUCTION. However, the crack growth rate is now one calculated by a least-squares fit of points determined from the sum of crack lengths of the two cracks. This calculation is described in Vol. II. If the time until failure is less than the remaining service life of the aircraft, the event FAILURE is scheduled. The time until the residual strength of the element reaches the fail-safe strength is calculated and event REACH.FAIL.SAFE.LGT is scheduled. This event can be scheduled in events ENTER.SERVICE, INSTALL.MODIFICATION, and REPAIR.

16.2 Local Variables

ARG - This real variable is used as an intermediate value in the calculation of time until element failure.

CL - This real variable is the crack length when the crack initiated.

GR1 - This real variable is the slow crack growth rate.

K10, K12, K2, K8, LGK5, N - These real variables are used as intermediate values in the calculation of time until element failure.

R2 - This real variable is the strength degradation rate when the composite crack length is between the critical crack length and the fail-safe length.

SF - This real variable is the fail-safe strength of the element.

SU - This real variable is the ultimate strength of the element.

T - This real variable is the time in flight hours until a crack initiated internally becomes external.

TAR - This real variable is the simulation time at which the aircraft being processed retires from service.

TCL - This real variable is the time it takes a single crack to grow from its initiation to the critical crack length.

T1 - This real variable is the time until the first crack reaches its critical crack length.

W, WXS, WY, Y - These real variables are used as intermediate values in the calculation of time until structural failure.

CCL - This real variable is the critical crack length of the element.

DL - This real variable is the length of the first crack at the time of corrosion initiation.

GR2 - This real variable is the fast crack growth rate of the element.

ID2SR - This integer variable is the aircraft identification number.

K1, K11, K13, K4, K9, LG - These real variables are used as intermediate values in the calculation of time until structural failure.

LIST - This real variable is set equal to 1.0 when the long list option is in effect and the aircraft being processed is one of those being tracked.

R1 - This real variable is the strength degradation rate between the time that a first crack initiates until the time that this crack reaches its critical length.

R3 - This real variable is the strength degradation rate from fail-safe strength until structural failure.

SMW - This real variable is used as an intermediate value in the calculation of time until structural failure.

S1 - This real variable is the element residual strength when the first crack is at the critical crack length.

TAC - This real variable is the time of corrosion initiation.

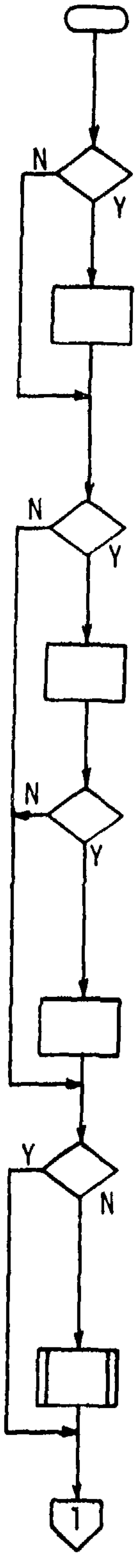
TA1 - This real variable is the time of first crack initiation.

TTF - This real variable is the time in flight hours until structural failure.

T2 - This real variable is the time in flight hours from when an element has a residual strength until when it has a fail-safe strength.

WX, WXY, X - These real variables are used as intermediate values in the calculation of time until structural failure.

16.3 Flow Chart



Event 2.STRENGTH.REDUCTION .

Is long list option in effect?

Print ID number and flight hours for aircraft being tracked.

Does crack initiate internally?

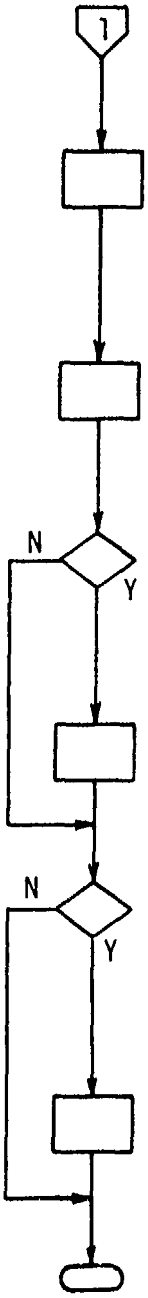
Set 2.INT to "YES".

Is time until crack becomes external less than remaining service life of aircraft?

Schedule event 2.ITE.

Is aircraft presently inspected below overhaul level?

Call routine INSPECTION.SCHEDULER.



Calculate time until residual strength equals fail-safe strength.

Calculate time until structural failure.

Is time until fail-safe strength less than remaining service life of aircraft?

Schedule event REACH.FAIL.SAFE.LGT.

Is time until failure less than remaining service life of aircraft?

Schedule event FAILURE.

Return.

17. 3.STRENGTH.REDUCTION

17.1 Description

This event represents the third crack initiation. If the long list option is in effect, the aircraft identification number and flight hours are printed for those aircraft being tracked. A uniform random number is compared with the probability of internal cracking to determine whether this crack initiates internally. If it does initiate internally, the time until it becomes external is calculated and the event 3.ITE is scheduled. Next, the time to structural failure is calculated by using the same three-part residual strength curve as in event 1.STRENGTH.REDUCTION. However, the crack growth rate is now calculated by a least-squares fit of points determined from the sum of crack lengths of the three cracks. This calculation is described in Vol. II. If the time until failure is less than the remaining service life of th aircraft, the event FAILURE is scheduled. The time until the residual strength of the element reaches the fail-safe strength is calculated and event REACH.FAIL.SAFE.LGT is scheduled. This event can be scheduled in events ENTER.SERVICE, INSTALL.MODIFICATION, and REPAIR.

17.2 Local Variables

ARG - This real variable is used as an intermediate value in the calculation of time until structural failure.

CL - This real variable is the crack length when the crack initiated.

GR1 - This real variable is the slow crack growth rate.

K10, K12, K2, K8, LGK5, N - These real variables are used as intermediate values in the calculation of time until structural failure.

R2 - This real variable is the strength degradation rate when the composite crack length is between the critical crack length and the fail-safe length.

SF - This real variable is the fail-safe strength of the element.

SU - This real variable is the ultimate strength of the element.

T - This real variable is the time in flight hours until a crack initiated internally becomes external.

TAR - This real variable is the simulation time at which the aircraft being processed retires from service.

TA2 - This real variable is the time of the second crack initiation.

TTF - This real variable is the time in flight hours until structural failure.

T2 - This real variable is the time in flight hours from when the element has a residual strength until it reaches a fail-safe strength.

WX, WXY, X, Y - These real variables are used as intermediate values in the calculation of time until structural failure.

1CL - This real variable is the crack length of the first crack at third crack initiation.

CCL - This real varaible is the critical crack length of the element.

DL - This real variable is the length of the first crack at the time of corrosion initiation.

GR2 - This real variable is the fast crack growth rate of the element.

ID3SR - This integer variable is the aircraft identification number.

K1, K11, K13, K4, K9, LG - These real variables are used as intermediate values in the calculation of time until structural failure.

LIST - This real variable is set equal to 1.0 when the long list option is in effect and the aircraft being processed is one of those being tracked.

R1 - This real variable is the strength degradation rate from when the first crack initiates until it reaches its critical crack length.

R3 - This real variable is the strength degradation rate from fail-safe strength until structural failure.

SMW - This real variable is used as an intermediate value in the calculation of time until structural failure.

S1 - This real variable is the element residual strength when the first crack is at the critical crack length.

TAC - This real variable is the time of corrosion initiation.

TA1 - This real variable is the time of first crack initiation.

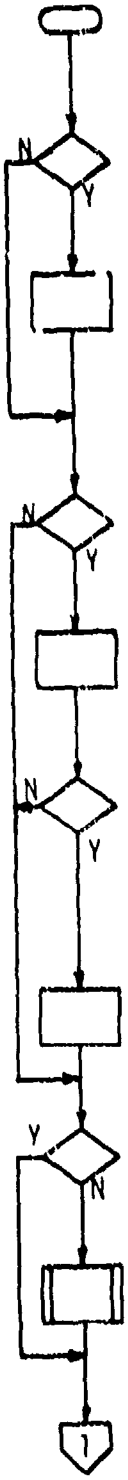
TCL - This real variable is the time it takes a single crack to grow from its initiation to the critical crack length.

T1 - This real variable is the time until the first crack reaches its critical crack length.

W, WXS, WY, XK, Y2 - These real variables are used as intermediate values in the calculation of time until structural failure.

2CL - This real variable is the length of the second crack at the time of the third crack initiation.

17.3 Flow Chart



Event 3.STRENGTH.REDUCTION .

Is long list option in effect?

Print ID number and flight hours for aircraft being tracked.

Does crack initiate internally?

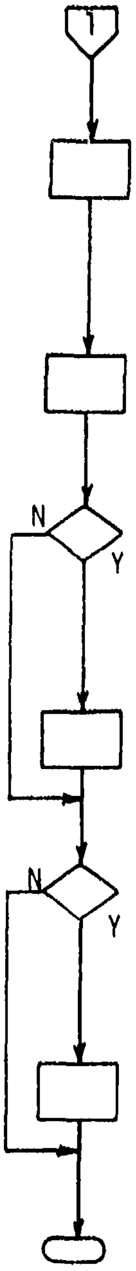
Set 3.INT to "YES".

Is time until crack becomes external less than remaining service life of aircraft?

Schedule event 3.ITE.

Is aircraft presently inspected below overhaul level?

Call routine INSPECTION.SCHEDULER.



Calculate time until residual strength equals fail-safe strength.

Calculate time until structural failure.

Is time until fail-safe strength less than remaining service life of aircraft?

Schedule event REACH.FAIL.SAFE.LGT.

Is time until failure less than remaining service life of aircraft?

Schedule event FAILURE.

Return.

18. 1.ITE

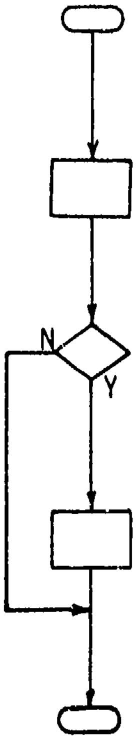
18.1 Description

This event represents the time when a first crack which initiated internally becomes external. This time is defined as the time when the element crack length reaches a percentage of the critical crack length. This percentage is an input parameter. At this time the appropriate element of the alpha array 1.INT is changed from "YES" to "NO". If the long list option is in effect, the aircraft identification and flight hours are printed for those aircraft being tracked. This event is scheduled in event 1.STRENGTH.REDUCTION.

18.2 Local Variables

ID1E - This integer variable is the aircraft identification number.

18.3 Flow Chart



Event 1.ITE.

Set 1.INT to "NO".

Is long list in effect?

Print A/C ID and flight hours of aircraft being tracked.

Return.

19. 2.ITE

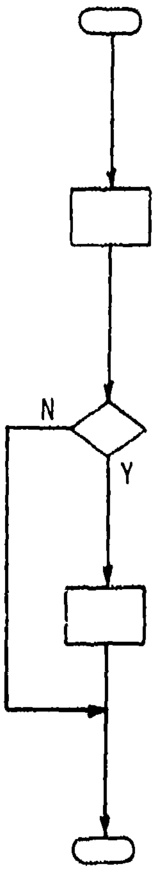
19.1 Description

This event represents the time when a second crack which initiated internally becomes external. This time is defined as the time when the element crack length reaches a percentage of the critical crack length. This percentage is an input parameter. At this time the appropriate element of the alpha array 2.INT is changed from "YES" to "NO". If the long list option is in effect, the aircraft identification and flight hours are printed for those aircraft being tracked. This event is scheduled in event 2.STRENGTH.REDUCTION.

19.2 Local Variables

ID2E - This integer variable is the aircraft identification number.

19.3 Flow Chart



Event 2.ITE.

Set 2.INT to "NO".

Is long list in effect?

Print A/C ID and flight hours of aircraft being tracked.

Return.

20. 3.ITE

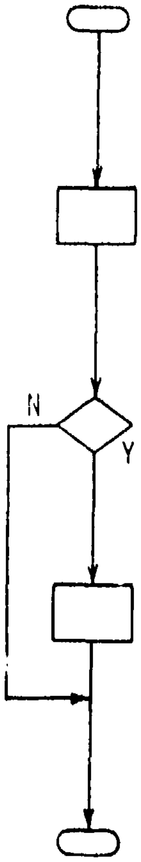
20.1 Description

This event represents the time when a first crack which initiated internally becomes external. This time is defined as the time when the element crack length reaches a percentage of the critical crack length. This percentage is an input parameter. At this time the appropriate element of the alpha array 3.INT is changed from "YES" to "NO". If the long list option is in effect, the aircraft identification and flight hours are printed for those aircraft being tracked. This event is scheduled in event 3.STRENGTH.REDUCTION.

20.2 Local Variables

ID3E - This integer variable is the aircraft identification number.

20.3 Flow Chart



Event 3.ITE.

Set 3.ITE to "NO".

Is long list in effect?

Print A/C ID and flight hours of aircraft being tracked.

Return.

21. INSPECTION.SCHEDULER

21.1 Description

This routine schedules all inspections below the overhaul level on a given aircraft. To conserve execution time, the inspections are scheduled so that the aircraft is not inspected before the defect reaches its minimum detectable size at each level of inspection. This routine is called from events 1. STRENGTH.REDUCTION and COROSION.

21.2 Local Variables

C1 - This real variable is the corrosion growth rate used to calculate the time to the minimum detectable corrosion area.

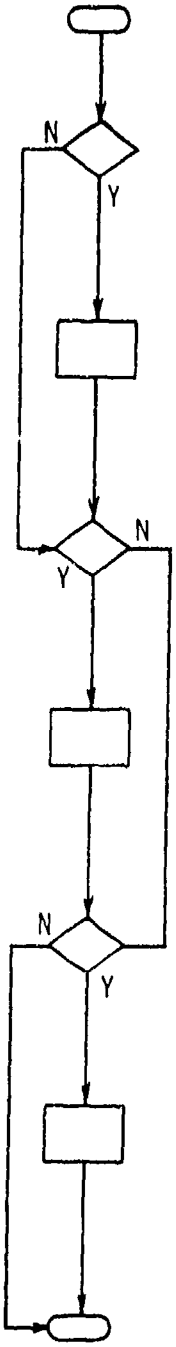
N - This integer variable indicates whether a crack initiation or a corrosion initiation caused this routine to be called.

TML - This real variable is the time to the minimum detectable defect size calculated for each level of inspection.

M1 - This real variable is the crack growth rate used to calculate the time to the minimum detectable crack length.

S.INSP.AT - This real variable is the simulation time at which the first inspection at each level is scheduled.

21.3 Flow Chart



Routine INSPECTION.SCHEDULER.

Is lowest external inspection level less than or equal to A-level?

Calculate time to minimum detectable size. Schedule A-level inspection.

Is lowest external inspection level less than or equal to B-level?

Calculate time to minimum detectable size. Schedule B-level inspection.

Is lowest external inspection level less than or equal to C-level?

Calculate time to minimum detectable size. Schedule C-level inspection.

Return.

22. A.LEVEL.INSPECTION

22.1 Description

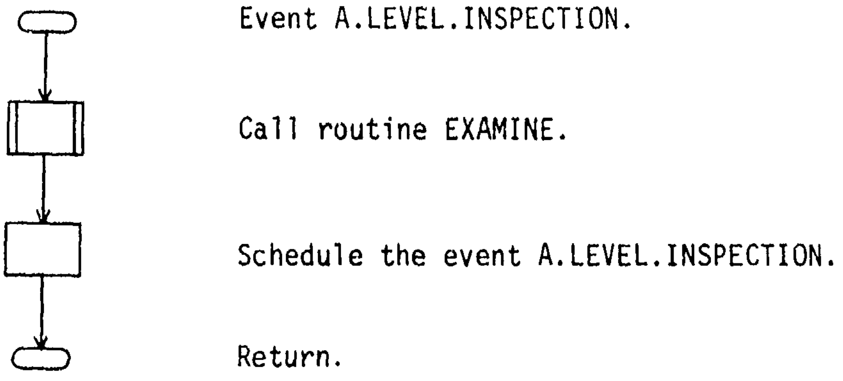
This event represents the performance of an A-level inspection. The constants which define the probability of detection equation at the A-level are passed to the routine EXAMINE which handles the actual inspection calculations for all levels of inspection. This event can be scheduled in the routine INSPECTION. SCHEDULER and the events A.LEVEL.INSPECTION, B.LEVEL.INSPECTION, C.LEVEL.INSPECTION, and D.LEVEL.INSPECTION.

22.2 Local Variables

FOUND - Not used in this event, this real variable is returned by routine EXAMINE with a non-zero value whenever a defect is found.

IDA - This integer variable is the identification number of the aircraft being inspected.

22.3 Flow Chart


Event A.LEVEL.INSPECTION.
Call routine EXAMINE.
Schedule the event A.LEVEL.INSPECTION.
Return.

23. B.LEVEL.INSPECTION

23.1 Description

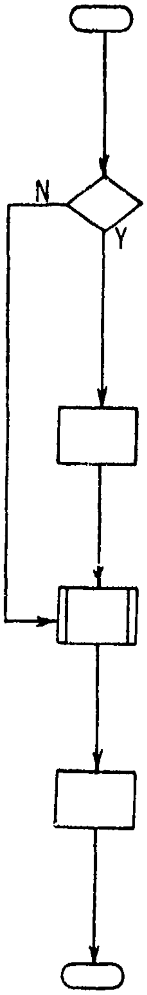
This event represents the performance of a B-level inspection. The constants which define the probability of detection equation at the B-level are passed to the routine EXAMINE. The inspection interval for each inspection level is not necessarily an even multiple of all lower level intervals. If an A-level inspection is scheduled, it is cancelled and rescheduled at present time plus one A-level interval later. This event can be scheduled in the routine INSPECTION.SCHEDULER and the events B.LEVEL.INSPECTION, C.LEVEL.INSPECTION, and D.LEVEL.INSPECTION.

23.2 Local Variables

FOUND - Not used in this event, this real variable is returned by routine EXAMINE with a non-zero value whenever a defect is found.

IDB - This integer variable is the identification number of the aircraft being inspected.

23.3 Flow Chart



Event B.LEVEL.INSPECTION.

Is lowest external inspection level less than or equal to A-level?

Reschedule the A-level inspection.

Call routine EXAMINE.

Schedule the event B.LEVEL.INSPECTION.

Return.

24. C.LEVEL.INSPECTION

24.1 Description

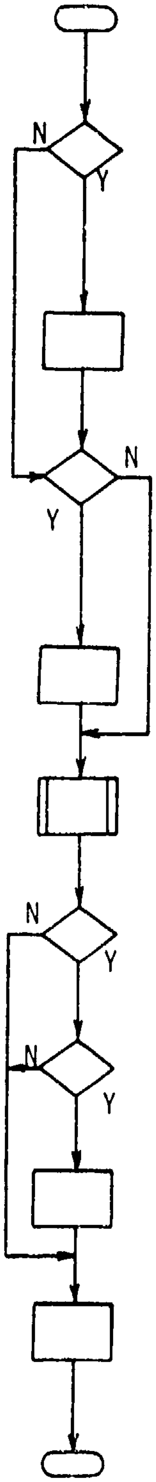
This event represents the performance of a C-level inspection. If there is either an A-level or a B-level inspection currently scheduled, it is cancelled and rescheduled at one A-level interval or B-level interval, respectively, later. The constants which define the probability of defect detection equation at the C-level are passed to the routine EXAMINE. If a crack is detected and the aircraft is one of the ten high-time aircraft, the time of detection is stored. This event can be scheduled in the routine INSPECTION.SCHEDULER and the events C.LEVEL.INSPECTION and D.LEVEL.INSPECTION.

24.2 Local Variables

FOUND - This real variable is returned by routine EXAMINE with a value of one whenever a crack is detected.

IDC - This integer variable is the identification number of the aircraft being inspected.

24.3 Flow Chart



Event C.LEVEL.INSPECTION.

Is lowest external inspection level less than or equal to B-level?

Reschedule B-level inspection.

Is lowest external inspection level less than or equal to A-level?

Reschedule A-level inspection.

Call routine EXAMINE.

Was crack detected?

Is aircraft one of ten high-time aircraft?

Retain time of detection.

Schedule event C.LEVEL.INSPECTION.

Return.

25. D.LEVEL.INSPECTION

25.1 Description

This event represents the performance of a D-level inspection. If the long list option is in effect on this element, the identification number and flight hours are printed for each aircraft being tracked. If there are any lower level inspections currently scheduled, they are cancelled and rescheduled at one inspection interval later. The constants which define the probability of defect detection equation at the D-level are passed to the routine EXAMINE. The defect histories of the ten high-time aircraft are now examined. If all of the ten high-time aircraft have gone one D-level interval without any cracks detected at either the C-level or the D-level, then the C-level and D-level inspection intervals are increased by the input factor FREQ. DECREASE.

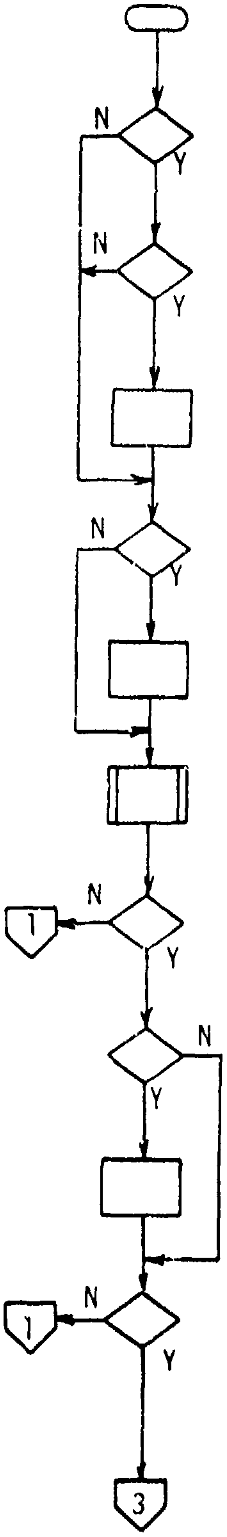
Any pending modifications are installed at this time. This event can be scheduled in the events ENTER.SERVICE and D.LEVEL. INSPECTION.

25.2 Local Variables

FOUND - This real variable is returned by routine EXAMINE with a value of one whenever a crack is detected.

IDD - This integer variable is the identification number of the aircraft being inspected.

25.3 Flow Chart



Event D.LEVEL.INSPECTION.

Is long list option in effect on this element?

Is this aircraft being tracked?

Print aircraft number and flight hours.

Are lower level inspections scheduled?

Reschedule lower level inspections.

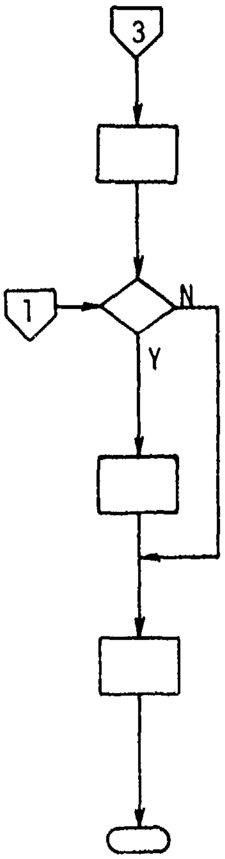
Call routine EXAMINE.

Is aircraft one of ten high-time aircraft?

Was crack detected?

Retain time of detection.

Have all ten high-time aircraft gone one D-level interval without anv cracks detected at either C-level or D-level?



Increase C-level and D-level inspection intervals.

Is a modification pending on this aircraft?

Install modification.

Schedule event D.LEVEL.INSPECTION.

Return.

26. EXAMINE

26.1 Description

This routine performs the numerical comparison which determines whether a defect is detected at each level of inspection. The constants which define the probability of defect detection equation are passed to this routine from the events which represent the different levels of inspection. If the long list option is in effect, each time a defect is detected on one of the aircraft being tracked, the following are printed: the size of the defect, the inspection level, the aircraft identification number, and the number of flight hours on the aircraft. This routine can be called from events A.LEVEL.INSPECTION, B.LEVEL.INSPECTION, C.LEVEL.INSPECTION, and D.LEVEL.INSPECTION.

26.2 Local Variables

AREA - This real variable is the calculated area of the corrosion defect.

CL - This real variable is the calculated length of the fatigue cracks.

LIST - This real variable is set equal to 1.0 if the long list option is in effect and the aircraft being inspected is one of those being tracked.

M2 - This real variable is the fast crack growth rate for the aircraft being inspected.

TAC - This real variable is the simulation time of the corrosion initiation.

TA2 - This real variable is the simulation time of the second crack initiation.

XA - This real variable is one of the probability of detection equation constants passed by the calling event.

YA - This real variable is one of the probability of detection equation constants passed by the calling event.

Z - This alpha variable is the level of inspection being performed.

ZL - This real variable is one of the probability of detection equation constants passed by the calling event.

CCL - This real variable is the critical crack length of the element.

FOUND - This real variable is set equal to two whenever corrosion is detected and to one whenever a crack is detected.

M1 - This real vairable is the slow crack growth rate for the aircraft being inspected.

N - This integer variable identifies the inspection level and is passed by the calling event.

TA1 - This real variable is the simulation time of the first crack initiation.

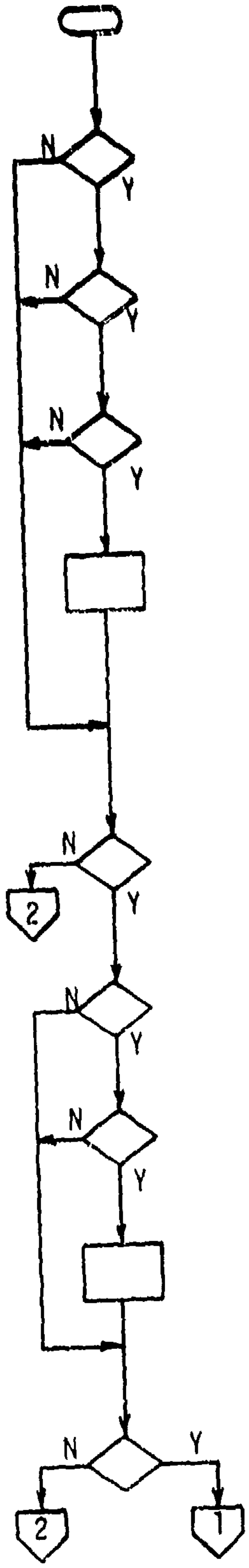
TA3 - This real variable is the simulation time of the third crack initiation.

XL - This real variable is one of the probability of detection equation constants passed by the calling event.

YL - This real variable is one of the probability of detection equation constants passed by the calling event.

ZA - This real variable is one of the probability of detection equation constants passed by the calling event.

20.3 Flow Chart



Routine EXAMINE.

Does corrosion exist?

Is the area inspected?

Is random number less than probability of detection?

Set tally counters to defect size.
Let FOUND=2.

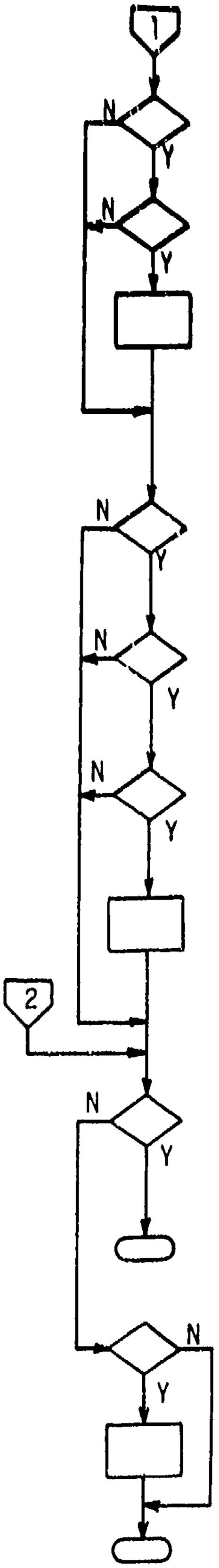
Does first crack exist?

Is the area inspected?

Is random number less than probability of detection?

Set tally counters to defect size.
Let FOUND=1.

Does second crack exist?



Is the area inspected?

Is random number less than probability of detection?

Set tally counters to defect size.
Let FOUND=1.

Does third crack exist?

Is the area inspected?

Is random number less than probability of detection?

Set tally counters to defect size.
Let FOUND=1.

Was routine called from D.LEVEL.INSPECTION?

Return.

Is FOUND greater than zero?

Schedule event REPAIR.

Return.

27. PODD

27.1 Description

This routine, defined as a function in the PREAMBLE, computes the probability of detecting a crack or corrosion defect of a given size. This probability is returned to the calling routine. PODD is called from routine EXAMINE and event IMMEDIATE.FLEET.INSPECTION.

27.2 Local Variables

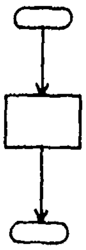
L - This real variable is the size of the defect under consideration.

Y - This real variable is an empirically determined equation constant for each level of inspection.

X - This real variable is the maximum probability of detection at a given inspection level.

Z - This real variable is the minimum defect size detectable at a given inspection level.

27.3 Flow Chart



Routine PODD.

Compute probability of detection.

Return.

28. CANCEL.SCHEDULED.INSPECTIONS

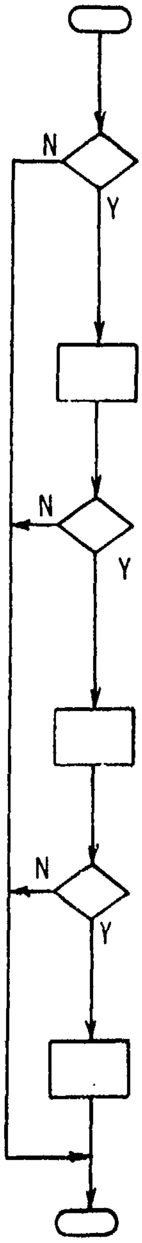
28.1 Description

This routine cancels all scheduled inspections below the overhaul level on a given aircraft. Whenever a defect is detected and repaired, it is assumed that all other defects existing on that particular element are also repaired. This routine is called at this time to cancel all subsequent inspections. Also, if an element fails or an aircraft with existing defects is retired, this routine is called to cancel all scheduled inspections. This routine can be called from events FAILURE, RETIRE. FROM.SERVICE, and REPAIR.

28.2 Local Variables

There are no local variables in this routine.

28.3 Flow Chart



Routine CANCEL.SCHEDULED.INSPECTIONS.

Is lowest level external inspection less than or equal to C-level?

Cancel the C-level inspection.

Is lowest level external inspection less than or equal to B-level?

Cancel the B-level inspection.

Is lowest level external inspection equal to A-level?

Cancel the A-level inspection.

Return.

29. REACH.FAIL.SAFE.LGT

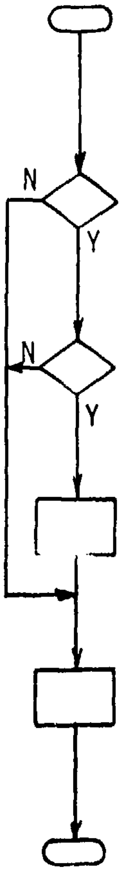
29.1 Description

This event represents the time when the residual strength of the element has been reduced to the fail-safe strength. The time and aircraft identification number are saved as part of the output. The calculation of the strength reduction is based on the sum of all crack lengths in the element. This event can be scheduled in events 1.STRENGTH.REDUCTION, 2.STRENGTH.REDUCTION, and 3.STRENGTH.REDUCTION.

29.2 Local Variables

IDRFS - This integer variable is the identification number of the aircraft being processed.

29.3 Flow Chart



Event REACH.FAIL.SAFE.LGT.

Is long list option in effect?

Is this aircraft one of those being tracked?

Print aircraft number and flight hours.

Build output arrays.

Return.

30. FAILURE

30.1 Description

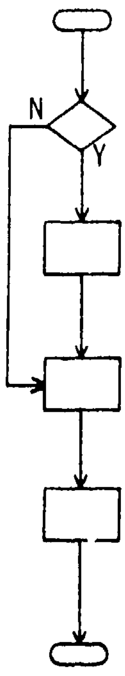
This event represents structural failure. When this event occurs, the aircraft is removed from the active fleet. If this aircraft was one of the ten high-time aircraft being monitored for the purpose of increasing inspection intervals, it is replaced by the next high-time aircraft in active service. Any remaining scheduled events are cancelled and their event notices destroyed. This event can be scheduled in events 1.STRENGTH. REDUCTION, 2.STRENGTH.REDUCTION, and 3.STRENGTH.REDUCTION.

30.2 Local Variables

HOLD - This integer variable serves as an intermediate storage for aircraft identification numbers when replacing one of the ten high-time aircraft.

IDFA - This integer variable is the identification number of the aircraft under consideration.

30.3 Flow Chart



Event FAILURE.

Is aircraft one of ten high time aircraft?

Replace with next high time aircraft in active service?

Remove aircraft from ACTIVE.FLEET.
File aircraft in CRASHED.FLEET.

Cancel scheduled events; destroy event notices.

Return.

31. RETIRE.FROM.SERVICE

31.1 Description

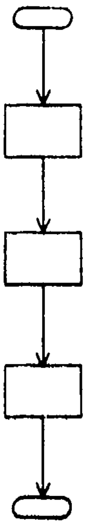
This event represents the retirement of an aircraft from active service. The aircraft is replaced in the HI.TIME.ACRFT array by that active aircraft not in the array which has the highest time in service. The aircraft being retired is removed from the set ACTIVE.FLEET and filed in the set FLEET.RETIRED. All remaining scheduled events for this aircraft are cancelled and the event notices destroyed. This event can only be scheduled in the event ENTER.SERVICE.

31.2 Local Variables

HOLD - This integer variable is used as an intermediate storage to hold aircraft identification numbers during the HI.TIME.ACRFT replacement.

IDRET - This integer variable is the identification number of the aircraft being retired from service.

31.3 Flow Chart



Event RETIRE.FROM.SERVICE.

Replace aircraft in HI.TIME.ACRFT array.

Remove aircraft from ACTIVE.FLEET, file in FLEET.RETIRED.

Cancel any scheduled events and destroy event notices.

Return.

32. REPAIR

32.1 Description

This event represents the structural repair of an element. If there is a modification pending on the element, it is installed at this time. If the events FAILURE and REACH.FAIL. SAFE.LGT are scheduled, they are cancelled. It is assumed that all existing defects are repaired and that new times to defect occurrences are determined in the same manner as when the aircraft entered service. The size and number of all existing cracks are compared with the inspection interval decrease criteria. If these criteria are met, then the events which decrease inspection intervals and perform special fleet-wide inspections are scheduled. All defects that were scheduled but had not occurred by this time are not affected by this event and will occur as originally scheduled. This event can be scheduled in events D.LEVEL.INSPECTION, EXAMINE, and IMMEDIATE.FLEET.INSPECTION.

32.2 Local Variables

AAFL - This real variable is the actual average fatigue life of the element design.

CL - This real variable is the calculated fatigue crack length.

HOURS.TO.CORROSION - This real variable is the time in flight hours to corrosion initiation.

MAX.CRK - This real variable is the maximum crack length in the element.

RST - This real variable is the remaining service time of the aircraft.

STR.RED - This real variable is the element strength reduction because of all existing cracks.

TA1 - This real variable is the simulation time of the first crack initiation.

TA3 - This real variable is the simulation time of the third crack initiation.

CCL - This real variable is the element critical crack length.

FIRST.LIFE - This real variable is the time in flight hours to first crack initiation.

IDREP - This integer variable is the aircraft identification number.

POT.CRK - This real variable is the sum of the maximum crack length and the length that the crack will grow during the shortest internal inspection interval.

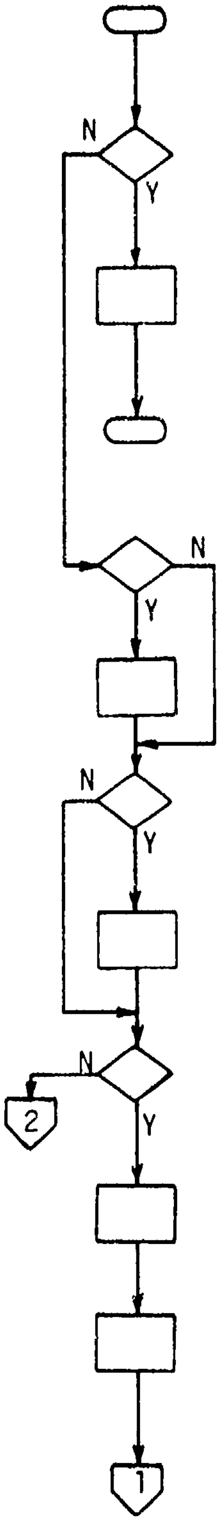
SECOND.LIFE - This real variable is the time in flight hours to the second crack initiation.

TAC - This real variable is the simulation time of the corrosion initiation.

TA2 - This real variable is the simulation time of the second crack initiation.

THIRD.LIFE - This real variable is the time in flight hours to the third crack initiation.

32.3 Flow Chart



Event REPAIR.

Is modification pending on this aircraft?

Install modification.

Return.

Is event FAILURE scheduled?

Cancel event FAILURE.

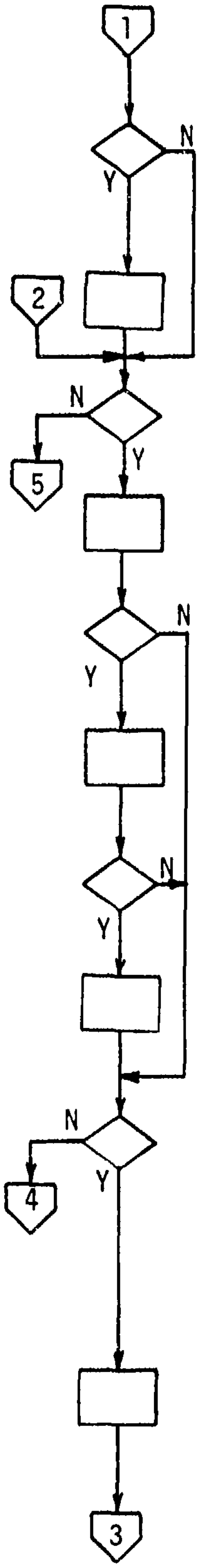
Is event REACH.FAIL.SAFE.LGT scheduled?

Cancel event REACH.FAIL.SAFE.LGT.

Does corrosion exist?

Set corrosion factor equal to 1.0.

Determine new time to corrosion initiation.



Is time to corrosion initiation less than remaining service life of aircraft?

Schedule corrosion initiation.

Does first crack exist?

Calculate crack length.

Does second crack exist?

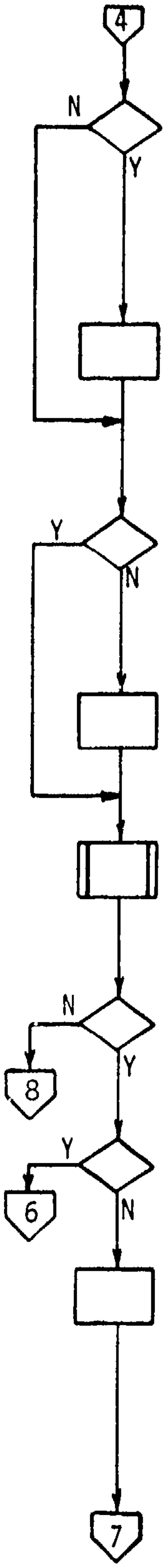
Calculate crack length.

Does third crack exist?

Calculate crack length.

Is sum of crack lengths plus crack growth rate times inspection interval greater than one-half the fail-safe crack length?

Increase inspection frequency and perform a special fleet wide inspection.



Is sum of crack lengths found in entire fleet greater than one-fifth of fleet size times fail-safe crack length?

Increase inspection frequency and perform a special fleet wide inspection.

Is there a modification pending somewhere in the fleet?

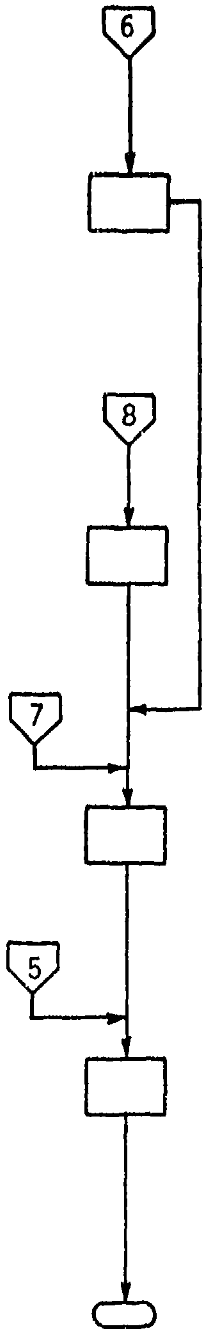
Schedule event DECISION.ON.MOD.

Call routine FATIGUE.LIFE.SCATTER.

Does second crack exist?

Does third crack exist?

Let time to first crack initiation be set to time to third crack initiation and times to second and third crack initiations be set to shortest times from routine FATIGUE.LIFE.SCATTER.



Let times to all three crack initiations be taken from FATIGUE.LIFE.SCATTER.

Set time to first crack to time to second crack. Set time to second crack to time to third crack. Set time to third crack from FATIGUE.LIFE.SCATTER.

Reschedule crack initiations for those cracks with times to crack initiation less than remaining service life of aircraft.

Cancel scheduled inspections below overhaul level.

Return.

33. T.INSPECTION:INCREASE

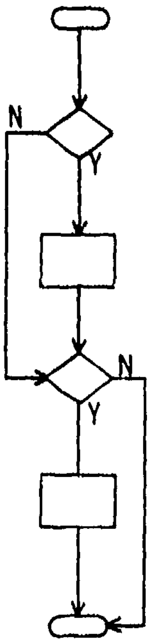
33.1 Description

This event represents an inspection frequency increase for a particular aircraft pending a structural modification because of a fatigue test failure. The factor by which the inspection frequency for both the close internal and the close external inspection is increased depends on the element being considered and is an input parameter. This event can only be scheduled in event T.IMPLEMENT.MOD.

33.2 Local Variables

IDTI - This integer variable is the identification number of the aircraft under consideration.

33.3 Flow Chart



Event T.INSPECTION.INCREASE.

Is lowest internal level or lowest external level of inspection equal to C-level?

Increase C-level inspection frequency.

Is lowest internal level or lowest external level of inspection equal to D-level?

Increase D-level inspection frequency.

Return.

34. INCREASE.INSPECTION.FREQUENCY

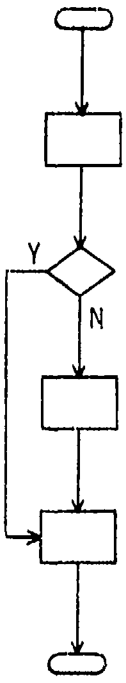
34.1 Description

This event represents a fleet-wide increase in the frequencies of the lowest level internal and external inspections. These may or may not be the same levels. However, the A-level and B-level frequencies are never changed. All scheduled inspections are cancelled and rescheduled to reflect the frequency increase. This event can be scheduled in events REPAIR and IMMEDIATE.FLEET.INSPECTION.

34.2 Local Variables

DIFF - This real variable is the difference in flight hours between the old and new lowest internal level inspection intervals.

34.3 Flow Chart



Event INCREASE.INSPECTION.FREQUENCY.

Increase lowest internal level frequency.

Is lowest internal level equal to lowest external level?

Increase lowest external level frequency.

Rescheduled scheduled inspections for each aircraft.

Returned.

35. IMMEDIATE.FLEET.INSPECTION

35.1 Description

This event represents an immediate fleet-wide inspection caused by finding a defect considered too hazardous to depend on scheduled inspections for detection of additional defects. Existing crack lengths and corrosion areas are calculated along with the associated probabilities of detection. As in the scheduled inspections, these probabilities are compared with a random number to determine whether or not the defect is detected. This event is always preceded by the event INCREASE.INSPECTION.FREQUENCY. Defects found during this inspection can cause an additional increase in the frequency of normally scheduled inspections. This event can only be scheduled in event REPAIR.

35.2 Local Variables

AREA - This real variable is the area in sq. inches of an existing corrosion defect.

CL - This real variable is the length in inches of an existing fatigue crack.

M1 - This real variable is the slow crack growth rate for a particular aircraft.

POT.CRK - This real variable is the sum of the maximum crack length detected on a particular aircraft plus the product of the average crack growth rate and the current lowest internal level inspection interval.

SR - This real variable is the sum of the lengths of all cracks detected on a particular aircraft.

TAC - This real variable is the time of initiation of an existing corrosion defect.

TA2 - This real variable is the time of initiation of an existing second fatigue crack.

CCL - This real variable is the critical crack length of the element under consideration.

FOUND - This real variable serves as a switch which is set equal to two when corrosion is detected and set equal to one when a crack is detected.

MAX.CRK - This real variable is the maximum crack length detected on a particular aircraft.

M2 - This real variable is the fast crack growth rate for a particular aircraft.

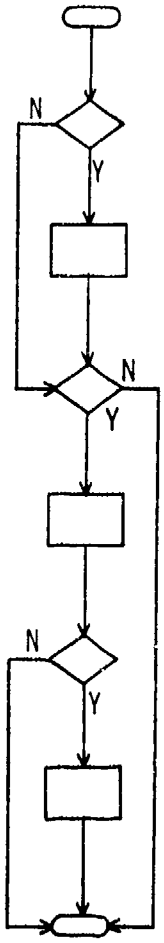
POT.STR.RED - This real variable is the sum of the detected crack lengths on a particular aircraft plus the product of the crack growth rate and the current lowest internal level inspection interval.

STR.RED - This real variable is equal to SR.

TA1 - This real variable is the time of initiation of an existing first fatigue crack.

TA3 - This real variable is the time of initiation of an existing third fatigue crack.

35.3 Flow Chart



Event IMMEDIATE.FLEET.INSPECTION.

Does corrosion exist?

Calculate corrosion area and test for detection.

Do cracks exist?

Calculate crack lengths and test for detection.

Do detected cracks meet criteria for inspection frequency increase?

Schedule event INCREASE.INSPECTION.FREQUENCY to occur immediately.

Return.

36. DECISION.ON.MOD

36.1 Description

This event makes the decision on whether or not to develop a structural modification because of service experience. The decision to develop a modification is made by comparing the cost per flight hour of the modification with the repair cost per flight hour plus the increased inspection cost per flight hour. The modification cost per flight hour is found by dividing the total fleet modification cost by the remaining service life of the fleet. The repair cost per flight hour is found by dividing the total fleet repair costs since the last modification by the fleet flight time since the last modification. The increased inspection cost per flight hour is found by dividing the projected increased inspection costs by the remaining service life of the fleet. A modification is justified when

$$MCPH < RCPH + ICPH$$

where MCPH = modification cost per flight hour
RCPH = repair cost per flight hour
ICPH = increased inspection cost per flight hour

This event can be scheduled only in event REPAIR.

36.2 Local Variables

ACCUMULATED.HRS - This real variable is the total fleet time since the last modification.

MD.COST - This real variable is the cost of installing a modification on a single aircraft. The costs for additional modifications can differ from those for the initial modifications.

NFTS - This real variable is the total production time of all aircraft entering service after the second production rate goes into affect.

POST.MOD.HRS - This real variable is the total fleet service time remaining after the modification.

TOOLING - This real variable is the tooling cost in the development of a modification. The costs for additional modifications can differ from those for the initial modifications.

ICPH - This real variable is the increased inspection cost per flight hour.

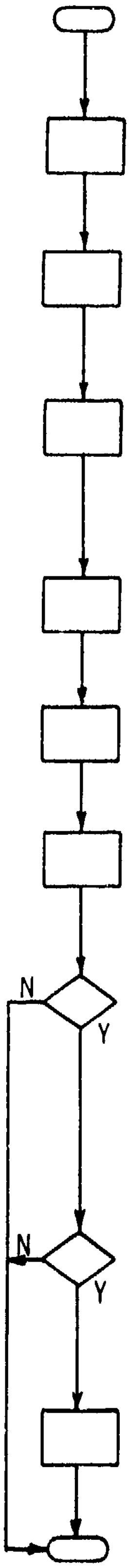
MCPH - This real variable is the modification cost per flight hour.

MRFH - This real variable is the service time remaining on a particular aircraft after its modification.

NPDL - This real variable is the number of aircraft which have entered service.

RCPH - This real variable is the repair cost per flight hour of the fleet.

36.3 Flow Chart



Event DECISION.ON.MOD.

Accumulate remaining service life of active fleet.

Determine fleet flight time since last modification.

Add service life of aircraft not yet produced to remaining service life of active fleet.

Calculate increased inspection costs per flight hour.

Calculate modification cost per flight hour.

Calculate repair cost per flight hour.

Is repair cost per hour plus increased inspection cost per hour greater than modification cost per hour?

Can modification be developed before last aircraft in fleet retires?

Schedule event IMPLEMENT.MODIFICATION.

Return.

37. IMPLEMENT.MODIFICATION

37.1 Description

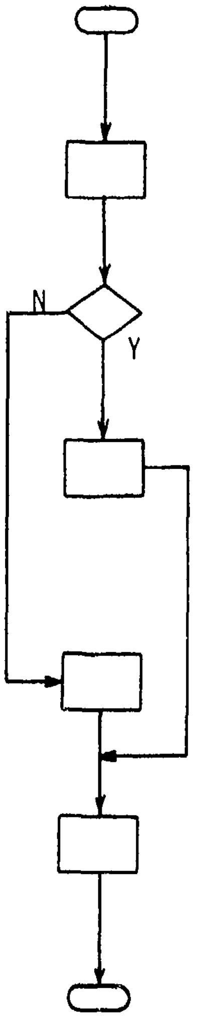
This event represents the development of a modification because of service experience. If the modification is to be fatigue tested, the actual average fatigue life is set equal to the original predicted life of the element design. Otherwise, the actual average fatigue life is determined by calling event REAL.LIFE. Elements of the alpha array SMOD.PENDING are set equal to "YES" to indicate which aircraft have service modifications pending. These modifications will be installed at the next D-level inspection or defect repair. Intervals for all levels of inspection are set to their initial values. This event can be scheduled only in event DECISION.ON.MOD.

37.2 Local Variables

NSIG - This real variable is the standard deviation of the ratio distribution passed to routine REAL.LIFE.

NMU - This real variable is the mean of the ratio distribution passed to routine REAL.LIFE.

37.3 Flow Chart



Event IMPLEMENT.MODIFICATION.

Set inspection intervals to initial values.

Is modification tested?

Set actual average fatigue life equal to predicted average fatigue life.

Call routine REAL.LIFE.

Set alpha array SMOD.PENDING.

Return.

38. DISPLAY.OUTPUT

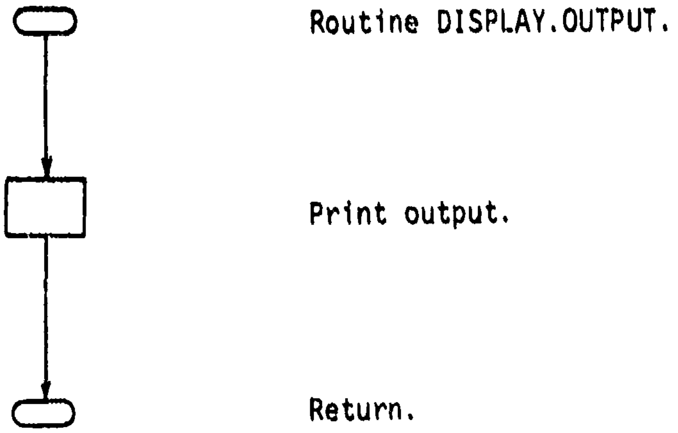
38.1 Description

This routine prints the standard output for each element. It is called from the MAIN program immediately after the completion of each element simulation. This output is suppressed if the long list option is in effect.

38.2 Local Variables

All the local variables are used to temporarily hold output values.

38.3 Flow Chart



39. SUMMARY

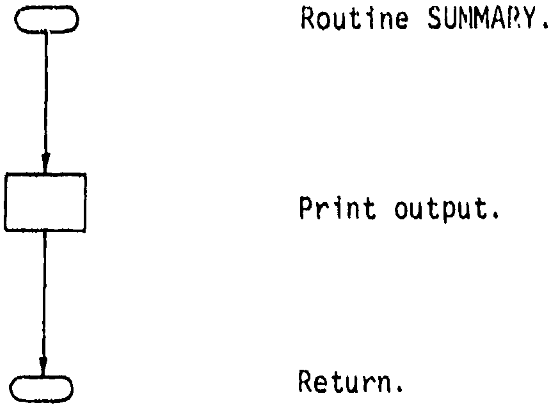
39.1 Description

This routine prints the standard output summary for each element type. The output of routine DISPLAY.OUTPUT for all the elements in an element type is contained in this output summary. This routine is called from the MAIN program each time a new element type is read in and at the end of the program run. This output is suppressed if the long list option is in effect.

39.2 Local Variables

All the local variables are used to temporarily hold output values.

39.3 Flow Chart



APPENDIX B

INITIAL PROGRAM SOURCE LISTING

```
>
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>> P L E A S E   N O T E <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<

LINE  CACI SIMSCRIPT II.5  RELEASE 8F                                  09/01/76  PAGE   1

PREAMBLE
        THE SYSTEM OWNS AN ACTIVE.FLEET, A FLEET.RETIRED AND A CRASHED.FLEET
''
'' GLOBAL REAL VARIABLES
''
        DEFINE PREDICTED.LIFE, ACTUAL.AVG.FAT.LIFE, START.TEST,
               TEST.ACCEL.FACT, BEGIN.PRODUCTION, PRODUCTION.TIME, USAGE.LIFE
            AS REAL VARIABLES
        DEFINE LGHT.TO.FAILURE, FLEET.STR.REQ, CRIT.CRK.LGT, LEAD.TIME, IAAFL
            AS REAL VARIABLES
        DEFINE HIGH.DEFECT.PROBABILITY, SUM, OCCURRENCE.RATE, C.GROWTH.RATE,
               CRRF, LOCATE.IN.STRESS.CC, SAMPLING,
               MU.R, SIG.R, ESAF.LGT,
               M1.MEAN, M2.MEAN, A, B, C7, C28, C29
            AS REAL VARIABLES
        DEFINE A.REPAIR.COST, B.REPAIR.COST,
               C.REPAIR.COST, D.REPAIR.COST, S.REPAIR.COST,
               COST.OF.REPAIRS, FIXIT.COST, 1ST.MOD.COST, AD.MOD.COST, 1ST.TOOLING,
               AD.TOOLING, I.FREQ.CHG, S.FREQ.CHG, CHG.FREQ.TIME
            AS REAL VARIABLES
        DEFINE I.PROB, C.PROB, ICRKT, ICONT, ISPT, ACRBL, BCRBL,
               CCRBL, DCRBL, SCRBL, ACA, BCA, CCA, DCA, SCA
            AS REAL VARIABLES
        DEFINE 1.CRK.OCCURRENCE.RATE, 2.CRK.OCCURRENCE.RATE, CRK.RATE.CHANGE,
               FREQ.DECREASE, 2.PRODUCTION.TIME, PRCHG, AIRFRAME.TIME
            AS REAL VARIABLES
        DEFINE GICRK, GICOR, GISO, GACRK, GBCRK, GCCRK, GDCRK, GSCRK, GACA, GBCA,
               GCCA, GDCA, GSCA, CINSL, BINSL, RCLI, KSMP
            AS REAL VARIABLES
''
'' GLOBAL INTEGER VARIABLES
''
        DEFINE ID, IDCR, I, SIZE.OF.FLEET, COUNT.ELEMENT, NICHG, LMTA,
               1.NUM.OF.RETIRE, 2.NUM.OF.CRASH, ITRNL, EXT.INSP.LEVEL,
               LIL, LFL, ID.NT.MODIFIED, PREV.MODIFIED, FDCR, CRR
            AS INTEGER VARIABLES
        DEFINE NICR, OCOR, OSOR, NPI, OSCR, OSCO, NSIC, NSMD, NSFL, NMD,
               NRES, SNRES, J, NNE, LOX,
               GNICR, GOCOR, GOSOR, GOSCR, GOSCO, GOPN, SNSIC, SNMD, SNSMD, SNSFL
            AS INTEGER VARIABLES
''
'' GLOBAL ALPHA VARIABLES
''
        DEFINE TES.FAILURE, NOT.TESTED,  PREVIOUSLY.MODIFIED, IMP.MOD.SCH,
               LONG.LIST, LIMIT,
               DEC.OR.MOD.SCH, SEL1, SEL2, SEL3, INT.LVL.INSP, EXT.LVL.INSP
            AS ALPHA VARIABLES
''
'' REAL ARRAYS
''
        DEFINE C.INTERVAL, D.INTERVAL, AHCD, CAHCD, IAHCD, CKREP.TIME, DKREP.TIME,
               LAST.SD, DEC.OR.MOD, ASR, MER, SC, SD, XMN, XMOD, CSRI
            AS REAL, 1-DIMENSIONAL ARRAYS
```

```
LINE  CACI SIMSCRIPT II.5  RELEASE 8F                              09/01/76  PAGE   2

      ''
      '' INTEGER ARRAYS
      ''
           DEFINE A1SR, A2SR, A3SR, AF, AIRPLANE, AAL, AHL, ACL, ADL, AC, ATII, ACID,
                  DICR, DICO, SACID, GDICR, GDICO, HI.TIME.ACRFT, APID, SAPID, STIM,
                  SSTIM, FLTHR, SFLTHR, ARFSL, NOAC, A1E, A2E, A3E
              AS INTEGER, 1-DIMENSIONAL ARRAYS
           DEFINE TLID AS AN INTEGER, 2-DIMENSIONAL ARRAY
      ''
      '' ALPHA ARRAYS
      ''
           DEFINE MODEL, ELEMENT, 1.CR.EXISTS, 2.CR.EXISTS, 3.CR.EXISTS, CO.EXISTS,
                  SP.SCH, SSTAN, SFLNR, AIL, FSH, IE1, IE2, IE3,
                  TMOD.PENDING, SMOD.PENDING, INSP.SCH, 1.INT, 2.INT, 3.INT, C.INT
              AS ALPHA, 1-DIMENSIONAL ARRAYS
           DEFINE ELID AS AN ALPHA, 2-DIMENSIONAL ARRAY
      ''
         TEMPORARY ENTITIES
      ''
           EVERY AIRCRAFT HAS AN ENTRY.TIME, A TAIL.ID
           AND MAY BELONG TO AN ACTIVE.FLEET, A FLEET.RETIRED AND A CRASHED.FLEET
           DEFINE TAIL.ID AS AN INTEGER VARIABLE
      ''
           TALLY NICR AS THE MINIMUM, XICR AS THE MAXIMUM, MICR AS THE MEAN OF ICRKT
           TALLY NICO AS THE MINIMUM, XICO AS THE MAXIMUM, MICO AS THE MEAN OF ICORT
           TALLY NISD AS THE MINIMUM, XISD AS THE MAXIMUM, MISD AS THE MEAN OF ISDT
           TALLY NACR AS THE MINIMUM, XACR AS THE MAXIMUM, MACR AS THE MEAN OF ACRKL
           TALLY NHCR AS THE MINIMUM, XHCR AS THE MAXIMUM, MHCR AS THE MEAN OF HCRKL
           TALLY NCCR AS THE MINIMUM, XCCR AS THE MAXIMUM, MCCR AS THE MEAN OF CCRKL
           TALLY NDCR AS THE MINIMUM, XDCR AS THE MAXIMUM, MDCR AS THE MEAN OF DCRKL
           TALLY NSCR AS THE MINIMUM, XSCR AS THE MAXIMUM, MSCR AS THE MEAN OF SCRKL
           TALLY NACA AS THE MINIMUM, XACA AS THE MAXIMUM, MACA AS THE MEAN OF ACA
           TALLY NHCA AS THE MINIMUM, XHCA AS THE MAXIMUM, MHCA AS THE MEAN OF HCA
           TALLY NCCA AS THE MINIMUM, XCCA AS THE MAXIMUM, MCCA AS THE MEAN OF CCA
           TALLY NDCA AS THE MINIMUM, XDCA AS THE MAXIMUM, MDCA AS THE MEAN OF DCA
           TALLY NSCA AS THE MINIMUM, XSCA AS THE MAXIMUM, MSCA AS THE MEAN OF SCA
           TALLY SICR AS THE MINIMUM, LICR AS THE MAXIMUM, AICR AS THE MEAN OF GICRK
           TALLY SICO AS THE MINIMUM, LICO AS THE MAXIMUM, AICO AS THE MEAN OF GICOR
           TALLY SISD AS THE MINIMUM, LISD AS THE MAXIMUM, AISD AS THE MEAN OF GISD
           TALLY SACR AS THE MINIMUM, LACR AS THE MAXIMUM, AACR AS THE MEAN OF GACRK
           TALLY SHCR AS THE MINIMUM, LHCR AS THE MAXIMUM, AHCR AS THE MEAN OF GHCRK
           TALLY SCCR AS THE MINIMUM, LCCR AS THE MAXIMUM, ACCR AS THE MEAN OF GCCRK
           TALLY SDCR AS THE MINIMUM, LDCR AS THE MAXIMUM, ADCR AS THE MEAN OF GDCRK
           TALLY SSCR AS THE MINIMUM, LSCR AS THE MAXIMUM, ASCR AS THE MEAN OF GSCRK
           TALLY SACA AS THE MINIMUM, LACA AS THE MAXIMUM, AACA AS THE MEAN OF GACA
           TALLY SHCA AS THE MINIMUM, LHCA AS THE MAXIMUM, AHCA AS THE MEAN OF GHCA
           TALLY SCCA AS THE MINIMUM, LCCA AS THE MAXIMUM, ACCA AS THE MEAN OF GCCA
           TALLY SOCA AS THE MINIMUM, LDCA AS THE MAXIMUM, ADCA AS THE MEAN OF GOCA
           TALLY SSCA AS THE MINIMUM, LSCA AS THE MAXIMUM, ASCA AS THE MEAN OF GSCA
           TALLY NCINS AS THE MINIMUM, XCINS AS THE MAXIMUM OF CINSL
           TALLY NDINS AS THE MINIMUM, XDINS AS THE MAXIMUM OF DINSL
      ''
         EVENT NOTICES INCLUDE ENTER.SERVICE, T.IMPLEMENT.MOD, IMPLEMENT.MODIFICATION,
''
       INCREASE.INSPECTION.FREQUENCY, IMMEDIATE.FLEET.INSPECTION,
       DECISION.ON.MOD
       EVERY IN.SERVICE.DAMAGE HAS AN IDSDM
            DEFINE IDSDM AS AN INTEGER VARIABLE
       EVERY CORROSION HAS AN IDCO
            DEFINE IDCO AS AN INTEGER VARIABLE
       EVERY 1.STRENGTH.REDUCTION HAS AN ID1SR
            DEFINE ID1SR AS AN INTEGER VARIABLE
       EVERY 2.STRENGTH.REDUCTION HAS AN ID2SR
            DEFINE ID2SR AS AN INTEGER VARIABLE
       EVERY 3.STRENGTH.REDUCTION HAS AN ID3SR
            DEFINE ID3SR AS AN INTEGER VARIABLE
       EVERY 1.ITE HAS AN ID1E
            DEFINE ID1E AS AN INTEGER VARIABLE
       EVERY 2.ITE HAS AN ID2E
            DEFINE ID2E AS AN INTEGER VARIABLE
       EVERY 3.ITE HAS AN ID3E
            DEFINE ID3E AS AN INTEGER VARIABLE
       EVERY A.LEVEL.INSPECTION HAS AN IDA
            DEFINE IDA AS AN INTEGER VARIABLE
       EVERY B.LEVEL.INSPECTION HAS AN IDB
            DEFINE IDB AS AN INTEGER VARIABLE
       EVERY C.LEVEL.INSPECTION HAS AN IDC
            DEFINE IDC AS AN INTEGER VARIABLE
       EVERY D.LEVEL.INSPECTION HAS AN IDD
            DEFINE IDD AS AN INTEGER VARIABLE
       EVERY T.INSPECTION.INCREASE HAS AN IDTI
            DEFINE IDTI AS AN INTEGER VARIABLE
       EVERY REPAIR HAS AN IDREP
            DEFINE IDREP AS AN INTEGER VARIABLE
       EVERY FAILURE HAS AN IDFA
            DEFINE IDFA AS AN INTEGER VARIABLE
       EVERY REACH.FAIL.SAFE.LGT HAS AN IDRFS
            DEFINE IDRFS AS AN INTEGER VARIABLE
       EVERY RETIRE.FROM.SERVICE HAS AN IDRET
            DEFINE IDRET AS AN INTEGER VARIABLE
''
    PRIORITY ORDER IS REPAIR, INCREASE.INSPECTION.FREQUENCY,
       IMMEDIATE.FLEET.INSPECTION, D.LEVEL.INSPECTION, C.LEVEL.INSPECTION,
       B.LEVEL.INSPECTION
''
       DEFINE PODD AS A REAL FUNCTION
       DEFINE RATE AS A REAL FUNCTION
''
       END
```

```
MAIN
''
'' SET RANDOM NUMBER SEEDS
''
        LET SEED.V(5) = SEED.V(6)
        LET SEED.V(6) = SEED.V(5)
''
'' RESERVE AIRCRAFT ARRAYS
''
        RESERVE HI.TIME.ACRFT(*), MMOD(*) AS 10
        RESERVE SC(*), SD(*) AS 30
        RESERVE ACID(*), FLTHR(*) AS 10
        RESERVE APID(*), STIM(*) AS 20
        RESERVE SACID(*), SFLTHR(*), SSTAN(*) AS 100
        RESERVE SAPID(*), SSTIM(*), SELNB(*) AS 200
        RESERVE MODEL(*) AS 2 AND FLEMENT(*) AS 4
        RESERVE ARCD(*), IARCD(*), CARCD(*) AS 4
''
'' READ AIRCRAFT INPUT
''
        READ MODEL(1), MODEL(2) AS 2 A 4
        READ SIZE.OF.FLEET, USAGE.LIFE
        READ BEGIN.PRODUCTION, PRODUCTION.TIME, 2.PRODUCTION.TIME, PRCHG
        READ START.TEST, TEST.ACCEL.FACT, C.GROWTH.RATE
        READ C7, C28, C29, MU.R, SIG.R, A, B
        READ IARCD(1), IARCD(2), IARCD(3), IARCD(4)
        READ CARCD(1), CARCD(2), CARCD(3), CARCD(4)
        READ SAMPLING, LONG.LIST, PCCL
      IF LONG.LIST = "YES"
        READ NOE
        RESERVE FLID(*,*) AS 4 BY NOE AND NOAC(*) AS NOE
        RESERVE TLID(*,*) AS NOE BY *
        FOR I = 1 TO NOE
        DO
        START NEW CARD
        READ FLID(1,I), ELID(2,I), FLID(3,I), FLID(4,I) AS 4 A 4
        READ NOAC(I)
        RESERVE TLID(I,*) AS NOAC(I)
        ALSO FOR J = 1 TO NOAC(I)
        DO
        READ TLID(I,J)
        LOOP
      ALWAYS
''
'' RESERVE ELEMENT ARRAYS
''
  RESERVE BICR(*), BICD(*), GUICR(*), GUICD(*) AS 4
  RESERVE AIRPLANE(*), TMOD.PENDING(*), SMOD.PENDING(*) AS SIZE.OF.FLEET
  RESERVE C.INTERVAL(*), AIL(*), ARFSL(*), FSH(*) AS SIZE.OF.FLEET
  RESERVE D.INTERVAL(*), OCCUR.MOD(*), CKREP.TIME(*) AS SIZE.OF.FLEET
  RESERVE CNREP.TIME(*), 1.CR.EXISTS(*), 2.CR.EXISTS(*) AS SIZE.OF.FLEET
  RESERVE 3.CR.EXISTS(*), CD.EXISTS(*), INSP.SCH(*), SD.SCH(*) AS SIZE.OF.FLEET
  RESERVE A1SR(*), A2SR(*), A3SR(*), AF(*), AC(*), ATII(*) AS SIZE.OF.FLEET
```

```
LINE  CACI SIMSCRIPT II.5  RELEASE 8F                    09/01/76  PAGE   5

     RESERVE AAL(*), ARL(*), ACL(*), ADL(*) AS SIZE.OF.FLEET
     RESERVE 1.INT(*), 2.INT(*), 3.INT(*), C.INT(*) AS SIZE.OF.FLEET
     RESERVE MSR(*), MFR(*), LAST.SD(*) AS SIZE.OF.FLEET
     RESERVE XRN(*), CGRI(*) AS SIZE.OF.FLEET
     RESERVE IE1(*), IE2(*), IE3(*) AS SIZE.OF.FLEET
     RESERVE A1E, A2E, A3E AS SIZE.OF.FLEET
   ''
   '' READ ELEMENT INPUT
   ''
   'READ' ADD 1 TO COUNT.ELEMENT
           START NEW CARD
           READ ELEMENT(1), ELEMENT(2), ELEMENT(3), ELEMENT(4) AS A A A
         IF ELEMENT(1) = "END "
           GO TO EOJ
         OTHERWISE
           READ PREDICTED.LIFE, ACTUAL.AVG.FAT.LIFE, M1.MEAN, M2.MEAN
           READ LGHT.I).FAILURE, CRIT.CRK.LGT, FSAF.LGT, BIRTH.DEFECT.PROBABILITY
           READ CRR, STR.OCCURRENCE.RATE, LEAD.TIME, 1.FREQ.CHG, S.FREQ.CHG
           READ FREQ.DECREASE, 1.PROB, C.PROB, INT.LVL.INSP, EXT.LVL.INSP
           READ A.REPAIR.COST, B.REPAIR.COST, C.REPAIR.COST, D.REPAIR.COST
           READ NUM.TESTED, 1ST.TOOLING, AD.TOOLING, 1ST.MD.COST, AD.MD.COST
           READ S.REPAIR.COST, LOCATED.IN.STRESS.CCA
           READ 1.COR.OCCURRENCE.RATE, 2.COR.OCCURRENCE.RATE, COR.RATE.CHANGE
   ''
   '' INITIALIZE GLOBAL VARIABLES
   ''
       IF SEL1 NE ELEMENT(1) OR SEL2 NE ELEMENT(2) OR SEL3 NE ELEMENT(3)
        IF COUNT.ELEMENT GT 1
         IF LONG.LIST = "NO"
          CALL SUMMARY
         ALWAYS
        REGARDLESS
          LET SEL1 = ELEMENT(1)
          LET SEL2 = ELEMENT(2)
          LET SEL3 = ELEMENT(3)
          CALL SUM.INITIALIZE
       REGARDLESS
   ''
          CALL INITIALIZATION
   ''
        IF LONG.LIST = "YES"
          FOR I = 1 TO NOF
          DO
         IF ELEMENT(1) = ELID(1,I) AND ELEMENT(2) = ELID(2,I) AND ELEMENT(3) =
          ELID(3,I) AND ELEMENT(4) = ELID(4,I)
          LET LINK = "YES"
          LET LOC = I
          LEAVE
         ELSE
          LOOP
        ALWAYS
        IF ACTUAL.AVG.FAT.LIFE = 0.0
          CALL REAL.LIFE(MU,R,SIG,W,PREDICTED.LIFE) YIELDING ACTUAL.AVG.FAT.LIFE
     REGARDLESS
       LET IAAFL = ACTUAL.AVG.FAT.LIFE
       LET NFTS = (SIZE.OF.FLEET + 1 - TRUNC.F(PHCHG/PRODUCTION.TIME)) *
       2.PRODUCTION.TIME
       LET SATL = START.TEST + IAAFL/TEST.ACCEL.FACT + LEAD.TIME
    IF SATL LT BEGIN.PRODUCTION + PHCHG + NFTS + USAGE.LIFE
     IF PREDICTED.LIFE LT 2.0 * USAGE.LIFE AND IAAFL LT PREDICTED.LIFE OR
       PREDICTED.LIFE GE 2.0 * USAGE.LIFE AND IAAFL LT 2.0 * USAGE.LIFE
       SCHEDULE A T.IMPLEMENT.MOD AT SATL
       LET TES.FAILURE = "YES"
     REGARDLESS
    REGARDLESS
     IF LIMO = "YES"
       START NEW PAGE
       WRITE MODEL(1), MODEL(2) AS S 50, "AIRCRAFT TYPE: ", 2 A 4
       SKIP 2 OUTPUT LINES
       PRINT 1 DOUBLE LINE WITH SIZE.OF.FLEET, USAGE.LIFE AS FOLLOWS
             NUMBER OF AIRCRAFT IN FLEET: *****                              AIRCRAFT SERVICE LIFE: ******* HOURS
       SKIP 1 OUTPUT LINE
       WRITE ELEMENT(1), ELEMENT(2), ELEMENT(3), ELEMENT(4)
       AS S 45, "STRUCTURAL ELEMENT: ", 4 A 4
       SKIP 2 OUTPUT LINES
      IF PREDICTED.LIFE = 0.0
       PRINT 1 DOUBLE LINE WITH IAAFL AS FOLLOWS
 ELEMENT DESIGN FATIGUE TESTED PRIOR TO PRODUCTION                           ACTUAL AVERAGE FATIGUE LIFE: ******* HOURS
       JUMP AHEAD
      ELSE
       PRINT 1 DOUBLE LINE WITH PREDICTED.LIFE, IAAFL AS FOLLOWS
         PREDICTED AVERAGE FATIGUE LIFE: ****** HOURS                    ACTUAL AVERAGE FATIGUE LIFE: ******* HOURS
      HERE
       SKIP 1 OUTPUT LINE
       PRINT 1 LINE AS FOLLOWS
                                              INITIAL INSPECTION INTERVALS
       SKIP 1 OUTPUT LINE
       PRINT 4 LINES WITH IAHCD(1), IAHCD(2), IAHCD(3), IAHCD(4) AS FOLLOWS
                                                 A-LEVEL ***** HOURS
                                                 B-LEVEL ***** HOURS
                                                 C-LEVEL ***** HOURS
                                                 D-LEVEL ***** HOURS
     ALWAYS
       SCHEDULE AN ENTER.SERVICE AT BEGIN.PRODUCTION
  ''
       START SIMULATION
      IF LONG.LIST = "NO"
       CALL DISPLAY.OUTPUT
      ALWAYS
       FOR EVERY AIRCRAFT IN CRASHED.FLEET
       DO
       REMOVE AIRPLANE(TAIL.ID) FROM CRASHED.FLEET
       DESTROY THE AIRCRAFT CALLED AIRPLANE(TAIL.ID)
       LOOP
       FOR EVERY AIRCRAFT IN FLEET.RETIRED
       DO
```

```
LINE  CACI SIMSCRIPT II.5  RELEASE 8F                              09/01/76  PAGE   7

 152        REMOVE AIRPLANE(TAIL.ID) FROM FLEET.RETIRED
 153        DESTROY THE AIRCRAFT CALLED AIRPLANE(TAIL.ID)
 154        LOOP
 155  ''
 156        GO READ
 157  'EOJ' IF LONG.LIST = "NO"
 158        CALL SUMMARY
 159        ALWAYS
 160        START NEW PAGE
 161        PRINT 1 LINE AS FOLLOWS
END OF SIMULATION
 162        STOP
 163        END


              LOCAL VARIABLES OF THIS ROUTINE

 I.1        INTEGER  WORD  3      I.2        INTEGER  WORD  4
 I.3        INTEGER  WORD  5      J.1        INTEGER  WORD  6
 K.1        INTEGER  WORD  8      K.2        INTEGER  WORD  9
 K.3        INTEGER  WORD 10      K.4        INTEGER  WORD 11
 L.11       INTEGER  WORD 18      L.13       INTEGER  WORD 19
 L.15       INTEGER  WORD 20      L.17       INTEGER  WORD 21
 L.22       INTEGER  WORD 22      L.26       INTEGER  WORD 24
 L.9        INTEGER  WORD 17      N.1        INTEGER  WORD  7
 NETS       DOUBLE   WORD 13      F.1        DOUBLE   WORD  1
 SATL       DOUBLE   WORD 15
```

```
ROUTINE INITIALIZATION
''
'' CHANGE  INSPECTION LEVEL CODE TO NUMERIC VALUE
''
         IF INT.LVL.INSP = "A"   LET ITRNL = 1       JUMP AHEAD
         OTHERWISE
         IF INT.LVL.INSP = "B"   LET ITRNL = 2       JUMP AHEAD
         OTHERWISE
         IF INT.LVL.INSP = "C"   LET ITRNL = 3       JUMP AHEAD
         OTHERWISE
                                 LET ITRNL = 4
      HERE
''
         IF EXT.LVL.INSP = "A"   LET EXT.INSP.LEVEL = 1   JUMP AHEAD
         OTHERWISE
         IF EXT.LVL.INSP = "B"   LET EXT.INSP.LEVEL = 2 · JUMP AHEAD
         OTHERWISE
         IF EXT.LVL.INSP = "C"   LET EXT.INSP.LEVEL = 3   JUMP AHEAD
         OTHERWISE
                                 LET EXT.INSP.LEVEL = 4
      HERE
''
'' CHANGE CORROSION RESISTANCE RATING TO MULTIPLYING FACTOR
''
         IF CRR = 1    LET CRRF = 1.50     JUMP AHEAD
         OTHERWISE
         IF CRR = 2    LET CRRF = 1.25     JUMP AHEAD
         OTHERWISE
         IF CRR = 3    LET CRRF = 1.00     JUMP AHEAD
         OTHERWISE
                       LET CRRF = 0.75
      HERE
         FOR I = 1 TO 4
         DO
         LET ARCD(I) = IARCD(I)
         LET OICR(I) = 0
         LET OICO(I) = 0
         LOOP
         FOR I = 1 TO 10
         DO
         LET HI.TIME.ACRFT(I) = I
         LET MRDD(I) = 0.0
         LOOP
''
         LET FOCK = 0
         LET LHTA = 10
         LET NICHG = 1
         LET OICH = 0
         LET OCOM = 0
         LET OSOM = 0
         LET OSCR = 0
         LET OSCO = 0
         LET OPD = 0
```

```
          LET NSFL = 0
          LET NRFS = 0
          LET NMD = 0
          LET NSMD = 0
          LET NSIC = 0
          LET 1.NUM.OF.RETIRE = 0
          LET 2.NUM.OF.CRASH = 0
          LET IOCK = 0
          LET TIME.V = 0.0
          LET AIRFRAME.TIME = 0.0
          LET COST.OF.REPAIRS = 0.0
          LET CINSL = IABCD(3)
          LET DINSL = IABCD(4)
          LET LIL = ITRNL
          LET LEL = EXT.INSP.LEVEL
        IF LEL = 1 OR LEL = 2
         IF LIL = 1 OR LIL = 2
          LET LIL = 3
         ALWAYS
          LET LEL = 3
        REGARDLESS
          LET T.FREQ.CHG = 1.0 - T.FREQ.CHG
          LET S.FREQ.CHG = 1.0 - S.FREQ.CHG
          LET FREQ.DECREASE = 1.0 + FREQ.DECREASE
          LET LTMD = "NO"
          LET TES.FAILURE = "NO"
          LET MOD.TESTED = "NO"
          LET DEC.ON.MOD.SCH = "NO"
          LET IMP.MOD.SCH = "NO"
          LET PREVIOUSLY.MODIFIED = "NO"
   ''
          FOR I = 1 TO SIZE.OF.FLEET
          DO
          LET IE1(I) = "NO"
          LET IE2(I) = "NO"
          LET IE3(I) = "NO"
          LET CGRI(I) = 1.0
          LET OCCUR.MOD(I) = 0.0
          LET AIL(I) = "NO"
          LET FSM(I) = "NO"
          LET INSP.SCH(I) = "NO"
          LET IMOD.PENDING(I) = "NO"
          LET C.INTERVAL(I) = IABCD(3)
          LET D.INTERVAL(I) = IABCD(4)
          LET SMOD.PENDING(I) = "NO"
          LET 1.CR.EXISTS(I) = "NO"
          LET 2.CR.EXISTS(I) = "NO"
          LET 3.CR.EXISTS(I) = "NO"
          LET CO.EXISTS(I) = "NO"
          LET SD.SCH(I) = "NO"
          LOOP
   ''
   '' RESET TALLY COUNTERS
```

```
      ''
              RESET TOTALS OF ICRKT
              RESET TOTALS OF ICORT
              RESET TOTALS OF ISDT
              RESET TOTALS OF ACRKL
              RESET TOTALS OF BCRKL
              RESET TOTALS OF CCRKL
              RESET TOTALS OF DCRKL
              RESET TOTALS OF SCRKL
              RESET TOTALS OF ACA
              RESET TOTALS OF BCA
              RESET TOTALS OF CCA
              RESET TOTALS OF DCA
              RESET TOTALS OF SCA
              RETURN
              END
```

```
ROUTINE SUM.INITIALIZE
        FOR I = 1 TO 4
        DO
        LET GUICHK(I) = 0
        LET GUICHW(I) = 0
        LOOP
        LET SNSTC = 0
        LET SNMD = 0
        LET SNSMC = 0
        LET SNSFL = 0
        LET SNMFS = 0
        LET GUICH = 0
        LET GOPO = 0
        LET GUCOW = 0
        LET GOSDM = 0
        LET GUSCR = 0
        LET GOSCO = 0
''
        RESET TOTALS OF GICHK
        RESET TOTALS OF GICHW
        RESET TOTALS OF GISO
        RESET TOTALS OF GACHK
        RESET TOTALS OF GMCHK
        RESET TOTALS OF GCCHK
        RESET TOTALS OF GDCHK
        RESET TOTALS OF GSCHK
        RESET TOTALS OF GACA
        RESET TOTALS OF GMCA
        RESET TOTALS OF GCCA
        RESET TOTALS OF GDCA
        RESET TOTALS OF GSCA
        RESET TOTALS OF CINSL
        RESET TOTALS OF DINSL
''
        RETURN
        END
```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
LINE  CACI SIMSCRIPT II.5  RELEASE 8F
                                                                            09/21/76  PAGE  12
   ROUTINE REAL.LIFE(MEAN,STD.DEV,POL) YIELDING RFL
   ''
   '' POL = PREDICTED AVERAGE FATIGUE LIFE OF ELEMENT DESIGN
   '' RFL = ACTUAL AVERAGE FATIGUE LIFE OF ELEMENT DESIGN
   '' RATIO = RATIO OF ACTUAL LIFE / PREDICTED LIFE
   '' IF RATIO > 10, RATIO IS SET TO 10
   '' IF RATIO < .1, RATIO IS SET TO .1
   '' DISTRIBUTION OF RATIOS IS REPRESENTED AS LOG NORMAL
   ''     MEAN = MEAN OF DISTRIBUTION OF RATIOS
   ''     STD.DEV = STANDARD DEVIATION OF DISTRIBUTION OF RATIOS

        LET RATIO = LOG.NORMAL.F(MEAN,STD.DEV,3)
        IF RATIO > 10.0            LET RATIO = 10.0          REGARDLESS
        IF RATIO < 0.1             LET RATIO =  0.1          REGARDLESS
        LET RFL = RATIO * POL
        RETURN
        END

              LOCAL VARIABLES OF THIS ROUTINE

MEAN        DOUBLE   WORD  1       POL         DOUBLE   WORD  5
RATIO       DOUBLE   WORD  9       RFL         DOUBLE   WORD  7
STD.DEV     DOUBLE   WORD  3
```

```
EVENT ENTER.SERVICE
        LET IDCK = IDCK + 1
        LET ID = IDCK
        CREATE AN AIRCRAFT CALLED AIRPLANE(ID)
        LET TAIL.ID(AIRPLANE(ID)) = ID
        LET ENTRY.TIME(AIRPLANE(ID)) = TIME.V
        FILE AIRPLANE(ID) IN ACTIVE.FLEET
        CALL FATIGUE.LIFE.SCATTER(ACTUAL.AVG.FAT.LIFE,8) YIELDING FIRST.LIFE,
        SECOND.LIFE AND THIRD.LIFE
        LET STD.FAST = M2.MEAN * .15
        LET STD.SLOW = M1.MEAN * .15
        LET MSR(ID) = RATE(M1.MEAN,STD.SLOW,5)
        LET MFR(ID) = RATE(M2.MEAN,STD.FAST,6)
        CALL PREDICT.CORROSION YIELDING HOURS.TO.CORROSION
        CALL PREDICT.SERVICE.DAMAGE YIELDING OURS.TO.SERVICE.DAMAGE
      IF HOURS.TO.CORROSION LT USAGE.LIFE
        SCHEDULE A COROSION(ID) AT TIME.V + HOURS.TO.CORROSION
        LET AC(ID) = COROSION
        LET COREP.TIME(ID) = TIME.V
        LET CO.EXISTS(ID) = "NS"
      REGARDLESS
 ''
      IF OURS.TO.SERVICE.DAMAGE LT USAGE.LIFE
        SCHEDULE AN IN.SERVICE.DAMAGE(ID) AT TIME.V + OURS.TO.SERVICE.DAMAGE
        LET LAST.SD(ID) = TIME.V
        LET SD.SCH(ID) = "YES"
      REGARDLESS
      IF RANDOM.F(7) LE BIRTH.DEFECT.PROBABILITY
        LET RN = RANDOM.F(3)
        LET DEFECT.LIFE = (-4042.6 * LOG.E.F(RN))**1.0132
        ADD 1 TO OPD
        ADD 1 TO SOPD
       IF DEFECT.LIFE LT FIRST.LIFE
        LET FIRST.LIFE = DEFECT.LIFE
       REGARDLESS
      REGARDLESS
     IF LTWO = "YES"
        FOR I = 1 TO NOAC(LOX)
        DO
      IF ID = TLID(LOX,I)
        LET LIST = 1.0
        SKIP 1 OUTPUT LINE
        PRINT 1 LINE WITH ID, TIME.V AS FOLLOWS
          A/C NO. *** ENTERS SERVICE ****** HOURS FROM START OF SIMULATION
       IF DEFECT.LIFE GT 0.0
        PRINT 1 LINE AS FOLLOWS
              PRODUCTION DEFECT PRESENT
       ALWAYS
        SKIP 1 OUTPUT LINE
        PRINT 3 LINES WITH FIRST.LIFE, SECOND.LIFE, THIRD.LIFE AS FOLLOWS
              1ST CRACK INITIATION PROJECTED AT ******* FLIGHT HOURS
              2ND CRACK INITIATION PROJECTED AT ******* FLIGHT HOURS
              3RD CRACK INITIATION PROJECTED AT ******* FLIGHT HOURS
        PRINT 2 LINES WITH MSR(ID), MFR(ID) AS FOLLOWS
              SLOW CRACK GROWTH RATE = .****** INCHES/HOUR
              FAST CRACK GROWTH RATE = .****** INCHES/HOUR
        LEAVE
      ELSE
        LOOP
     ALWAYS
 ''
     IF FIRST.LIFE LT USAGE.LIFE OR CO.EXISTS(ID)="NS" OR SD.SCH(ID)="YES"
        SCHEDULE A 1.STRENGTH.REDUCTION(ID) AT TIME.V + FIRST.LIFE
        LET A1SR(ID) = 1.STRENGTH.REDUCTION
        LET CKREP.TIME(ID) = TIME.V
        LET 1.CR.EXISTS(ID) = "NS"
      IF SECOND.LIFE LT USAGE.LIFE OR CO.EXISTS(ID)="NS" OR SD.SCH(ID)="YES"
        SCHEDULE A 2.STRENGTH.REDUCTION(ID) AT TIME.V + SECOND.LIFE
        LET A2SR(ID) = 2.STRENGTH.REDUCTION
        LET 2.CR.EXISTS(ID) = "NS"
       IF THIRD.LIFE LT USAGE.LIFE OR CO.EXISTS(ID)="NS" OR SD.SCH(ID)="YES"
        SCHEDULE A 3.STRENGTH.REDUCTION(ID) AT TIME.V + THIRD.LIFE
        LET A3SR(ID) = 3.STRENGTH.REDUCTION
        LET 3.CR.EXISTS(ID) = "NS"
       REGARDLESS
      REGARDLESS
     REGARDLESS
        SCHEDULE A D.LEVEL.INSPECTION(ID) AT TIME.V + D.INTERVAL(ID)
        LET ADL(ID) = D.LEVEL.INSPECTION
        SCHEDULE A RETIRE.FROM.SERVICE(ID) AT TIME.V + USAGE.LIFE
      IF IOCK EQ SIZE.OF.FLEET
        RETURN
      OTHERWISE
      IF TIME.V - BEGIN.PRODUCTION GE PRCHG
        SCHEDULE AN ENTER.SERVICE AT TIME.V + 2.PRODUCTION.TIME
        RETURN
      OTHERWISE
        SCHEDULE AN ENTER.SERVICE AT TIME.V + PRODUCTION.TIME
        RETURN
        END
```

```
                    LOCAL VARIABLES OF THIS ROUTINE

DEFECT.LIF DOUBLE    WORD 17       FIRST.LIFE DOUBLE    WORD  1
HOURS.TO.C DOUBLE    WORD 11       I.1        INTEGER   WORD 25
I.2        INTEGER   WORD 26       I.3        INTEGER   WORD 27
J.1        INTEGER   WORD 28       K.1        INTEGER   WORD 30
K.2        INTEGER   WORD 31       K.3        INTEGER   WORD 32
K.4        INTEGER   WORD 33       L.4        INTEGER   WORD 21
L.6        INTEGER   WORD 34       LIST       DOUBLE    WORD 19
N.1        INTEGER   WORD 29       OURS.TO.SF DOUBLE    WORD 13
R.1        DOUBLE    WORD 23       RN         DOUBLE    WORD 15
SECOND.LIF DOUBLE    WORD  3       STD.FAST   DOUBLE    WORD  7
STD.SLOW   DOUBLE    WORD  9       THIRD.LIFE DOUBLE    WORD  5
```

```
LINE  CACI SIMSCRIPT II.5  RELEASE 8F                                09/01/76  PAGE  15

    ROUTINE FATIGUE.LIFE.SCATTER(REL,N) YIELDING FIRST.LIFE, SECOND.LIFE AND
          THIRD.LIFE
    ''
    '' GENERATES ELEMENT FATIGUE LIVES REFLECTING BASIC FATIGUE SCATTER AND LOAD
    ''     ENVIRONMENT VARIATION
    ''
    '' DISTRIBUTION OF FATIGUE LIVES IS REPRESENTED AS TWO PARAMETER WEIBULL
    ''     ALPHA = SHAPE PARAMETER
    ''     BETA  = SCALE PARAMETER (CHARACTERISTIC VALUE)
    ''     N = RANDOM NUMBER STREAM
    ''
           DEFINE N AS AN INTEGER VARIABLE
           DEFINE LIFE AS A REAL, 1-DIMENSIONAL ARRAY
           RESERVE LIFE(*) AS 3
           LET ALPHA = 2.78
           LET BETA = REL/.89018
           LET RN = RANDOM.F(N)
           FOR I = 1 TO 3, DO
           LET LIFE(I) = BETA * (LOG.E.F(1/RN))**(1/ALPHA)
           LET RN = UNIFORM.F(0.0,RN,3)
           LOOP
           LET FIRST.LIFE = LIFE(1)
           LET SECOND.LIFE = LIFE(2)
           LET THIRD.LIFE = LIFE(3)
           RELEASE LIFE(*)
           RETURN
           END

               LOCAL VARIABLES OF THIS ROUTINE

 ALPHA      DOUBLE   WORD 13      BETA       DOUBLE   WORD 15
 FIRST.LIFE DOUBLE   WORD  5      LIFE       REAL     WORD 11
 N          INTEGER  WORD  3      REL        DOUBLE   WORD  1
 RN         DOUBLE   WORD 17      SECOND.LIF DOUBLE   WORD  7
 THIRD.LIFE DOUBLE   WORD  9
```

```
ROUTINE INSTALL.MODIFICATION
''
'' THIS ROUTINE REPRESENTS THE INSTALLATION OF A STRUCTURAL MODIFICATION CAUSED
''    BY A FATIGUE TEST FAILURE OR BY A SERVICE DEFECT FOUND IN THE ELEMENT
''
      LET NSMD = NSMD + 1
      LET BEEN.MODIFIED = BEEN.MODIFIED + 1
      LET C.INTERVAL(ID) = IARCD(3)
      LET D.INTERVAL(ID) = IARCD(4)
      LET OCCUR.MOD(ID) = TIME.V
''
'' CANCEL SCHEDULED INSPECTIONS
''
    IF INSP.SCH(ID) = "YES"
      CALL CANCEL.SCHEDULED.INSPECTIONS
    REGARDLESS
''
    IF FSH(ID) = "YES"
      LET FSH(ID) = "NO"
      LET FAILURE = AF(ID)
      CANCEL THE FAILURE
      DESTROY THE FAILURE
    ALWAYS
      CALL FATIGUE.LIFE.SCATTER(ACTUAL.AVG.FAT.LIFE,4) YIELDING FIRST.LIFE,
      SECOND.LIFE AND THIRD.LIFE
      LET STD.SLOW = M1.MEAN * .15
      LET STD.FAST = M2.MEAN * .15
      LET MSR(ID) = RATE(M1.MEAN,STD.SLOW,5)
      LET MFR(ID) = RATE(M2.MEAN,STD.FAST,6)
      LET RST = USAGE.LIFE - TIME.V + ENTRY.TIME(AIRPLANE(ID))
''
'' CANCEL PREVIOUSLY SCHEDULED CRACK AND CORROSION INITIATIONS, RESCHEDULE
''    THESE EVENTS IF THEY OCCUR WITHIN THE REMAINING SERVICE LIFE (RST) OF THE
''    AIRCRAFT
''
     IF CO.EXISTS(ID) NE "NN"
       LET CORROSION = AC(ID)
      IF CO.EXISTS(ID) = "NS"
       CANCEL THE CORROSION
      ALWAYS
       DESTROY THE CORROSION
       LET CO.EXISTS(ID) = "NN"
       LET CGRI(ID) = 1.0
     REGARDLESS
       CALL PREDICT.CORROSION YIELDING HOURS.TO.CORROSION
      IF HOURS.TO.CORROSION LT RST
       SCHEDULE A CORROSION(ID) AT TIME.V + HOURS.TO.CORROSION
       LET AC(ID) = CORROSION
       LET CORRP.TIME(ID) = TIME.V
       LET CO.EXISTS(ID) = "NS"
      REGARDLESS
''
     IF RANDOM.F(7) LE BIRTH.DEFECT.PROBABILITY
```

```
      LET OPD = OPD + 1
      LET RN = RANDOM.F(3)
      LFT DEFFCT.LIFF = (-4042.6 * LOG.E.F(RN))**1.0132
     IF DEFECT.LIFF LT FIRST.LIFE
      LET FIRST.LIFE = DEFECT.LIFE
     REGARDLFSS
    REGARDLESS
''
  IF 1.CR.EXISTS(ID) NF "NN"
      LFT 1.STRENGTH.REDUCTION = A1SR(ID)
    IF 1.CR.FXISIS(ID) = "NS"
      CANCFL THE 1.STRENGTH.REDUCTION
    REGARDLFSS
      DESTROY THE 1.STRFNGTH.REDUCTION
      LFT 1.CR.FXISTS(ID) = "NN"
   IF 2.CR.FXISTS(ID) NF "NN"
      LET 2.STRENGTH.REDUCTION = A2SR(ID)
    IF 2.CR.EXISTS(ID) = "NS"
      CANCEL THE 2.STRENGTH.REDUCTION
    RFGARDLESS
      DESTROY THE 2.STRFNGTH.REDUCTION
      LET 2.CR.EXISTS(ID) = "NN"
    IF 3.CR.FXISTS(ID) NF "NN"
      LET 3.STRENGTH.REDUCTION = A3SR(ID)
     IF 3.CR.EXISTS(ID) = "NS"
      CANCEL THE 3.STRENGTH.REDUCTION
     REGARDLESS
      DESTROY THE 3.STRFNGTH.REDUCTION
      LET 3.CR.EXISTS(ID) = "NN"
    RFGARDLESS
   REGARDLESS
  REGARDLESS
   IF IE1(ID) = "YES"
      LET 1.ITE = A1E(ID)
      LET IE1(ID) = "NO"
      CANCEL THE 1.ITF
      DESTROY THE 1.ITE
    IF IE2(ID) = "YES"
      LET 2.ITE = A2F(ID)
      LET IE2(ID) = "NO"
      CANCFL THE 2.ITE
      DESTROY THE 2.ITF
     IF IE3(ID) = "YES"
      LET 3.ITE = A3E(ID)
      LFT IE3(ID) = "NO"
      CANCEL THE 3.ITF
      DESTROY THE 3.ITE
     ALWAYS
    ALWAYS
   ALWAYS
    IF AIL(ID) = "YES"
      LET REACH.FAIL.SAFE.LGT = ARFSL(ID)
      CANCEL THE REACH.FAIL.SAFF.LGT
```

```
LINE  CACI SIMSCRIPT II.5  RELFASE 8F                              09/01/76  PAGE  18

          DESTROY THE REACH.FAIL.SAFE.LGT
          LET AIL(ID) = "NO"
        REGARDLESS
   ''
      IF FIRST.LIFE LT RST OR CO.EXISTS(ID) = "NS" OR SD.SCH(ID) = "YES"
          SCHEDULE A 1.STRENGTH.REDUCTION(ID) AT TIME.V + FIRST.LIFE
          LET A1SR(ID) = 1.STRENGTH.REDUCTION
          LET CKREP.TIME(ID) = TIME.V
          LET 1.CR.EXISTS(ID) = "NS"
       IF SECOND.LIFE LT RST OR CO.EXISTS(ID) = "NS" OR SD.SCH(ID) = "YES"
          SCHEDULE A 2.STRENGTH.REDUCTION(ID) AT TIME.V + SECOND.LIFE
          LET A2SR(ID) = 2.STRENGTH.REDUCTION
          LET 2.CR.EXISTS(ID) = "NS"
        IF THIRD.LIFE LT RST OR CO.EXISTS(ID) = "NS" OR SD.SCH(ID) = "YES"
          SCHEDULE A 3.STRENGTH.REDUCTION(ID) AT TIME.V + THIRD.LIFE
          LET A3SR(ID) = 3.STRENGTH.REDUCTION
          LET 3.CR.EXISTS(ID) = "NS"
        REGARDLESS
       REGARDLESS
      REGARDLESS
   ''
       IF LTHO = "YES"
          FOR I = 1 TO NOAC(LOX)
          DO
        IF ID = TLID(LOX,I)
          SKIP 1 OUTPUT LINE
          PRINT 6 LINES WITH ID, TIME.V-ENTRY.TIME(AIRPLANE(ID)), FIRST.LIFE,
          SECOND.LIFE, THIRD.LIFE, MSR(ID), MFR(ID) AS FOLLOWS
            MODIFICATION INSTALLED ON A/C NO. *** AT ****** FLIGHT HOURS
                 1ST CRACK INITIATION PROJECTED AT ******* FLIGHT HOURS
                 2ND CRACK INITIATION PROJECTED AT ******* FLIGHT HOURS
                 3RD CRACK INITIATION PROJECTED AT ******* FLIGHT HOURS
                 SLOW CRACK GROWTH RATE = .****** INCHES/HOUR
                 FAST CRACK GROWTH RATE = .****** INCHES/HOUR
          LEAVE
        ELSE
          LOOP
       ALWAYS
          RETURN
          END

                  LOCAL VARIABLES OF THIS ROUTINE

DEFECT.LIF DOUBLE    WORD 17       FIRST.LIFE DOUBLE    WORD  1
HOURS.TO.C DOUBLE    WORD 13       I.1        INTEGER   WORD 23
I.2        INTEGER   WORD 24       I.3        INTEGER   WORD 25
J.1        INTEGER   WORD 26       K.1        INTEGER   WORD 28
K.2        INTEGER   WORD 29       K.3        INTEGER   WORD 30
K.4        INTEGER   WORD 31       L.4        INTEGER   WORD 19
N.1        INTEGER   WORD 27       R.1        DOUBLE    WORD 21
RN         DOUBLE    WORD 15       RST        DOUBLE    WORD 11
SECOND.LIF DOUBLE    WORD  3       STD.FAST   DOUBLE    WORD  9
```

```
LINE  CACI SIMSCRIPT II.5  RELEASE 8F                        09/01/76  PAGE  19
 STD.SLOW    DOUBLE    WORD  7       THIRD.LIFE DOUBLE    WORD  5
```

```
EVENT IN.SERVICE.DAMAGE(IDSDM)
''
'' REPRESENTS THE OCCURRENCE OF A SERVICE DAMAGE DEFECT; RESULTS IN IMMEDIATE
''     INITIATION OF NEXT SCHEDULED CRACK
''
        DEFINE IDSDM AS AN INTEGER VARIABLE
        LET ID = IDSDM
    IF LTHO = "YES"
        FOR I = 1 TO NOAC(LOX)
        DO
      IF ID = ILID(LOX,I)
        LET LIST = 1.0
        SKIP 1 OUTPUT LINE
        PRINT 2 LINES WITH ID, TIME.V-ENTRY.TIME(AIRPLANE(ID)) AS FOLLOWS
          A/C NO. *** EXPERIENCES SERVICE DAMAGE AT ****** FLIGHT HOURS
               CAUSES IMMEDIATE INITIATION OF NEXT SCHEDULED CRACK
        LEAVE
      ELSE
        LOOP
     ALWAYS
        LET OSDM = OSDM + 1
        LET GOSDM = GOSDM + 1
        LET ISDT = TIME.V - LAST.SD(ID)
        LET GISD = TIME.V - LAST.SD(ID)
        CALL PREEDICT.SERVICE.DAMAGE YIELDING OURS.TO.SERVICE.DAMAGE
        LET RSI = USAGE.LIFE - TIME.V + ENTRY.TIME(AIRPLANE(ID))
        LET SD.SCH(ID) = "NO"
      IF OURS.TO.SERVICE.DAMAGE LT RSI
        SCHEDULE AN IN.SERVICE.DAMAGE(ID) AT TIME.V + OURS.TO.SERVICE.DAMAGE
        LET LAST.SD(ID) = TIME.V
        LET SD.SCH(ID) = "YES"
      REGARDLESS
''
'' TEST FOR SCHEDULED FIRST CRACK INITIATION
''
        LET 1.STRENGTH.REDUCTION = A1SR(ID)
        LET 2.STRENGTH.REDUCTION = A2SR(ID)
        LET 3.STRENGTH.REDUCTION = A3SR(ID)
      IF 1.CR.EXISTS(ID) = "NS"
        CANCEL THE 1.STRENGTH.REDUCTION
        RESCHEDULE THE 1.STRENGTH.REDUCTION(ID) NOW
       IF SD.SCH(ID) = "NO" AND TIME.A(2.STRENGTH.REDUCTION) GE RSI + TIME.V
        AND CO.EXISTS(ID) NE "NS"
        CANCEL THE 2.STRENGTH.REDUCTION
        DESTROY THE 2.STRENGTH.REDUCTION
        CANCEL THE 3.STRENGTH.REDUCTION
        DESTROY THE 3.STRENGTH.REDUCTION
        LET 2.CR.EXISTS(ID) = "NN"
        LET 3.CR.EXISTS(ID) = "NN"
       ALWAYS
        RETURN
      OTHERWISE
''
```

```
LINE  CACI SIMSCRIPT II.5  RELEASE 8F                                  09/01/76  PAGE  21

     '' TEST FOR SCHEDULED SECOND CRACK INITIATION
     ''
          IF 2.CR.EXISTS(ID) = "NS"
            CANCEL THE 2.STRENGTH.REDUCTION
            RESCHEDULE THE 2.STRENGTH.REDUCTION(ID) NOW
           IF S3.SCH(ID) = "NO" AND TIME.A(3.STRENGTH.REDUCTION) GE RST + TIME.V
            AND CO.EXISTS(ID) NE "NS"
            CANCEL THE 3.STRENGTH.REDUCTION
            DESTROY THE 3.STRENGTH.REDUCTION
            LET 3.CR.EXISTS(ID) = "ON"
           ALWAYS
            RETURN
         OTHERWISE
     ''
     '' TEST FOR SCHEDULED THIRD CRACK INITIATION
     ''
          IF 3.CR.EXISTS(ID) = "NS"
            CANCEL THE 3.STRENGTH.REDUCTION
            RESCHEDULE THE 3.STRENGTH.REDUCTION(ID) NOW
            RETURN
          OTHERWISE
            RETURN
            END

                   LOCAL VARIABLES OF THIS ROUTINE

  I.1          INTEGER   WORD  9       I.2          INTEGER  WORD 10
  I.3          INTEGER   WORD 11       IOSDM        INTEGER  WORD  1
  J.1          INTEGER   WORD 12       K.1          INTEGER  WORD 14
  K.2          INTEGER   WORD 15       K.5          INTEGER  WORD 16
  K.3          INTEGER   WORD 17       L.4          INTEGER  WORD  5
  LIST         DOUBLE    WORD  3       N.1          INTEGER  WORD 13
  OURS.TO.5H   DOUBLE    WORD 19       R.1          DOUBLE   WORD  7
  RST          DOUBLE    WORD 21
```

```
EVENT T.IMPLEMENT.MOD
''
'' REPRESENTS DEVELOPMENT OF MODIFICATION BECAUSE OF FATIGUE TEST FAILURE
''
      LET NMD = NMD + 1
      LET FOCK = IOCK
      LET BEEN.MODIFIED = 0
      LET TO.BE.MODIFIED = IOCK
      LET COST.OF.REPAIRS = 0.0
      LET AIRFRAME.TIME = 0.0
      LET ACTUAL.AVG.FAT.LIFE = PREDICTED.LIFE
    IF MOD.TESTED = "NO"
      LET NMU = MU.R + ((1.0-MU.R)*.15)
      LET NSIG = SIG.R * .85
      CALL REAL.LIFE(NMU,NSIG,PREDICTED.LIFE) YIELDING ACTUAL.AVG.FAT.LIFE
    REGARDLESS
      FOR EVERY AIRCRAFT IN ACTIVE.FLEET
      DO
      LET ID = TAIL.ID
    IF IAAFL LT USAGE.LIFE
      LET TMOD.PENDING(ID) = "YES"
     IF ENTRY.TIME(AIRPLANE(ID)) + C7*IAAFL GT TIME.V
      SCHEDULE A T.INSPECTION.INCREASE(ID) AT ENTRY.TIME(AIRPLANE(ID)) + C7 *
      IAAFL
      LET ATII(ID) = T.INSPECTION.INCREASE
     ALWAYS
    REGARDLESS
      LOOP
      RETURN
      END

              LOCAL VARIABLES OF THIS ROUTINE

L.4         INTEGER   WORD  5      NMU        DOUBLE   WORD  1
NSIG        DOUBLE    WORD  4
```

```
LINE  CACI SIMSCRIPT II.5  RELEASE 9F                          09/01/76  PAGE  23

      ROUTINE PREDICT.SERVICE.DAMAGE YIELDING HOURS.TO.SERVICE.DAMAGE
      ''
      '' GENERATES TIME TO SERVICE DAMAGE OCCURRENCE FOR A GIVEN AIRCRAFT FROM A
      ''    CONSTANT SERVICE DAMAGE OCCURRENCE RATE
      ''
          IF SDM.OCCURRENCE.RATE LE 0.0
            LET HOURS.TO.SERVICE.DAMAGE = 2.0 * USAGE.LIFE
            RETURN
          ELSE
            LET RN = RANDOM.F(2)
            LET HOURS.TO.SERVICE.DAMAGE = -LOG.E.F(RN) / SDM.OCCURRENCE.RATE
            RETURN
            END

               LOCAL VARIABLES OF THIS ROUTINE

 HOURS.TO.SE DOUBLE    WORD  1       RN          DOUBLE    WORD  3
```

```
LINE  CACI SIMSCRIPT II.5  RELEASE 8F                              09/01/76  PAGE  24

      ROUTINE PREDICT.CORROSION YIELDING HOURS.TO.CORROSION
      ''
      '' GENERATES TIME TO CORROSION INITIATION FROM ELEMENT HAZARD RATE WHICH IS
      ''     APPROXIMATED BY TWO CONSTANT OCCURRENCE RATES
      ''
      ''     1.COM.OCCURRENCE.RATE = FIRST CONSTANT OCCURRENCE RATE
      ''     2.COM.OCCURRENCE.RATE = SECOND CONSTANT OCCURRENCE RATE
      ''     COM.RATE.CHANGE = TIME ON AIRFRAME AT WHICH SECOND RATE IS USED
      ''
             LET RN = RANDOM.F(4)
             LET CRCT = COM.RATE.CHANGE - TIME.V + ENTRY.TIME(AIRPLANE(ID))
           IF CRCT LE 0.0
            IF 2.COM.OCCURRENCE.RATE LE 0.0
             LET HOURS.TO.CORROSION = 2.0 * USAGE.LIFE
             RETURN
            ELSE
             LET HOURS.TO.CORROSION = LOG.E.F(RN) / (-2.COM.OCCURRENCE.RATE)
             RETURN
           OTHERWISE
      ''
           IF RN GE EXP.F(-CRCT * 1.COM.OCCURRENCE.RATE)
            IF 1.COM.OCCURRENCE.RATE LE 0.0
             LET HOURS.TO.CORROSION = 2.0 * USAGE.LIFE
             RETURN
            ELSE
             LET HOURS.TO.CORROSION = LOG.E.F(RN) / (-1.COM.OCCURRENCE.RATE)
             RETURN
           OTHERWISE
            IF 2.COM.OCCURRENCE.RATE LE 0.0
             LET HOURS.TO.CORROSION = 2.0 * USAGE.LIFE
             RETURN
            ELSE
             LET LD = (2.COM.OCCURRENCE.RATE - 1.COM.OCCURRENCE.RATE) * CRCT
             LET HOURS.TO.CORROSION = LOG.E.F(RN/EXP.F(LD)) / (-2.COM.OCCURRENCE.RATE)
             RETURN
             END

                 LOCAL VARIABLES OF THIS ROUTINE

 CRCT          DOUBLE    WORD  5       HOURS.TO.C DOUBLE    WORD  1
 LD            DOUBLE    WORD  7       RN         DOUBLE    WORD  5
```

```
EVENT CORROSION(IOCO) SAVING THE EVENT NOTICE
''
'' REPRESENTS THE INITIATION OF CORROSION; REMAINING TIME TO CRACK INITIATION OF
''     ALL SCHEDULED CRACKS IS REDUCED BY COR.MULTIPLYING.FACTOR
''
       DEFINE IOCO AS AN INTEGER VARIABLE
       LET ID = IOCO
       LET RST = USAGE.LIFE - TIME.V + ENTRY.TIME(AIRPLANE(ID))
       LET TCORR = TIME.V - CORRP.TIME(ID)
       LET GTCORR = TIME.V - CORRP.TIME(ID)
       LET NCORR = NCORR + 1
       LET GNCORR = GNCORR + 1
       LET C.INI(ID) = "NO"
     IF RANDOM.F(10) LE C.PROB
       LET C.INI(ID) = "YES"
     REGARDLESS
       LET C1.EXISTS(ID) = "YES"
     IF INSP.SCH(ID) = "NO"
       CALL INSPECTION.SCHEDULER(2)
     REGARDLESS
       LET COR.MULTIPLYING.FACTOR = 1.0 - C29
     IF RANDOM.F(1) LE LOCATED.IN.STRESS.COR
       LET COR.MULTIPLYING.FACTOR = 1.0 - C28
     REGARDLESS
       LET CGRI(ID) = 2.0 - COR.MULTIPLYING.FACTOR
    IF LTHO = "YES"
       FOR I = 1 TO NMAC(LOX)
       DO
     IF ID = TLID(LOX,I)
       LET LIST = 1.0
       SKIP 1 OUTPUT LINE
       PRINT 1 LINE WITH ID, TIME.V-ENTRY.TIME(AIRPLANE(ID)) AS FOLLOWS
         A/C *1, *** EXPERIENCES CORROSION INITIATION AT ****** HOURS
       SKIP 1 OUTPUT LINE
       PRINT 2 LINES WITH VSR(ID)*CGRI(ID), VFR(ID)*CGRI(ID) AS FOLLOWS
              SLOW CRACK GROWTH RATE = .****** INCHES/HOUR
              FAST CRACK GROWTH RATE = .****** INCHES/HOUR
       LEAVE
     ELSE
       LOOP
    ALWAYS
   IF IE1(ID) = "YES"
       LET 1.ITE = A1E(ID)
       LET TRI = (TIME.A(1.ITE)-TIME.V) / CGRI(ID)
       CANCEL THE 1.ITE
      IF TRI LT RST
       RESCHEDULE THE 1.ITE(ID) AT TIME.V + TRI
       JUMP AHEAD
      ELSE
       DESTROY THE 1.ITE
       LET IE1(ID) = "NO"
      HERE
    IF IE2(ID) = "YES"
```

```
          LET 2.ITE = A2E(ID)
          LET TRT = (TIME.A(2.ITE)-TIME.V) / CGRI(ID)
          CANCEL THE 2.ITE
         IF TRT LT RST
          RESCHEDULE THE 2.ITE(ID) AT TIME.V + TRT
          JUMP AHEAD
         ELSE
          DESTROY THE 2.ITE
          LET IE2(ID) = "NO"
         HERE
        IF IE3(ID) = "YES"
         LET 3.ITE = A3E(ID)
         LET TRT = (TIME.A(3.ITE)-TIME.V) / CGRI(ID)
         CANCEL THE 3.ITE
        IF TRT LT RST
         RESCHEDULE THE 3.ITE(ID) AT TIME.V + TRT
         JUMP AHEAD
        ELSE
         DESTROY THE 3.ITE
         LET IE3(ID) = "NO"
        HERE
       ALWAYS
      ALWAYS
     ALWAYS
      IF AIL(ID) = "YES"
       LET REACH.FAIL.SAFE.LGT = ARFSL(ID)
       LET TRT = (TIME.A(REACH.FAIL.SAFE.LGT) - TIME.V) / CGRI(ID)
       CANCEL THE REACH.FAIL.SAFE.LGT
      IF TRT LT RST
       RESCHEDULE THE REACH.FAIL.SAFE.LGT(ID) AT TIME.V + TRT
       JUMP AHEAD
      ELSE
       DESTROY THE REACH.FAIL.SAFE.LGT
       LET AIL(ID) = "NO"
      HERE
     REGARDLESS
     IF FSH(ID) = "YES"
      LET FAILURE = AF(ID)
      LET NFTM = TIME.A(FAILURE) - TIME.V
     IF LIST = 1.0
      PRINT 1 LINE WITH TIME.V+NFTM*CON.MULTIPLYING.FACTOR-
      ENTRY.TIME(AIRPLANE(ID)) AS FOLLOWS
            ELEMENT FAILURE PROJECTED AT ****** FLIGHT HOURS
     ALWAYS
      CANCEL THE FAILURE
     IF NFTM LT RST
      RESCHEDULE THE FAILURE(ID) AT TIME.V + NFTM / CGRI(ID)
      JUMP AHEAD
     ELSE
      DESTROY THE FAILURE
      LET FSH(ID) = "NO"
     HERE
    REGARDLESS
```

```
LINE  CACI SIMSCRIPT II.5  RELEASE 8F                              09/01/76  PAGE  27

      ''
      '' TEST FOR SCHEDULED FIRST CRACK INITIATION
      ''
           IF 1.CR.EXISTS(ID) = "NS"
             LET 1.STRENGTH.REDUCTION = A1SR(ID)
             LET REMAINING.LIFE = TIME.A(1.STRENGTH.REDUCTION) - TIME.V
             LET REDUCED.REMAINING.LIFE = REMAINING.LIFE * CDR.MULTIPLYING.FACTOR
            IF LIST = 1.0
             PRINT 1 LINE WITH TIME.V+REDUCED.REMAINING.LIFE-ENTRY.TIME(AIRPLANE(ID))
             AS FOLLOWS
                    1ST CRACK INITIATION PROJECTED AT ****** FLIGHT HOURS
            ALWAYS
             CANCEL THE 1.STRENGTH.REDUCTION
             DESTROY THE 1.STRENGTH.REDUCTION
             LET 1.CR.EXISTS(ID) = "NO"
            IF REDUCED.REMAINING.LIFE LT RST OR SP.SCH(ID) = "YES"
             SCHEDULE A 1.STRENGTH.REDUCTION(ID) AT TIME.V + REDUCED.REMAINING.LIFE
             LET A1SR(ID) = 1.STRENGTH.REDUCTION
             LET 1.CR.EXISTS(ID) = "NS"
            ALWAYS
           REGARDLESS
      ''
      '' TEST FOR SCHEDULED SECOND CRACK INITIATION
      ''
           IF 2.CR.EXISTS(ID) = "NS"
             LET 2.STRENGTH.REDUCTION = A2SR(ID)
             LET REMAINING.LIFE = TIME.A(2.STRENGTH.REDUCTION) - TIME.V
             LET REDUCED.REMAINING.LIFE = REMAINING.LIFE * CDR.MULTIPLYING.FACTOR
            IF LIST = 1.0
             PRINT 1 LINE WITH TIME.V+REDUCED.REMAINING.LIFE-ENTRY.TIME(AIRPLANE(ID))
             AS FOLLOWS
                    2ND CRACK INITIATION PROJECTED AT ****** FLIGHT HOURS
            ALWAYS
             CANCEL THE 2.STRENGTH.REDUCTION
             DESTROY THE 2.STRENGTH.REDUCTION
             LET 2.CR.EXISTS(ID) = " "
            IF REDUCED.REMAINING.LIFE LT RST OR SP.SCH(ID) = "YES"
             SCHEDULE A 2.STRENGTH.REDUCTION(ID) AT TIME.V + REDUCED.REMAINING.LIFE
             LET A2SR(ID) = 2.STRENGTH.REDUCTION
             LET 2.CR.EXISTS(ID) = "NS"
            ALWAYS
           REGARDLESS
      ''
      '' TEST FOR SCHEDULED THIRD CRACK INITIATION
      ''
           IF 3.CR.EXISTS(ID) = "NS"
             LET 3.STRENGTH.REDUCTION = A3SR(ID)
             LET REMAINING.LIFE = TIME.A(3.STRENGTH.REDUCTION) - TIME.V
             LET REDUCED.REMAINING.LIFE = REMAINING.LIFE * CDR.MULTIPLYING.FACTOR
            IF LIST = 1.0
             PRINT 1 LINE WITH TIME.V+REDUCED.REMAINING.LIFE-ENTRY.TIME(AIRPLANE(ID))
             AS FOLLOWS
                    3RD CRACK INITIATION PROJECTED AT ****** FLIGHT HOURS
```

```
 153        ALWAYS
 154         CANCEL THE 3.STRENGTH.REDUCTION
 155         DESTROY THE 3.STRENGTH.REDUCTION
 156         LET 3.CH.EXISTS(ID) = "NO"
 157        IF REDUCED.REMAINING.LIFE LT HST OR SP.SCH(ID) = "YES"
 158         SCHEDULE A 3.STRENGTH.REDUCTION(ID) AT TIME.V + REDUCED.REMAINING.LIFE
 159         LET ASSN(ID) = 3.STRENGTH.REDUCTION
 160         LET 3.CH.EXISTS(ID) = "YES"
 161        ALWAYS
 162       REGARDLESS
 163        RETURN
 164        END


              LOCAL VARIABLES OF THIS ROUTINE
CUM.MULTIP DOUBLE   WORD  5
I.2        INTEGER  WORD 14      I.1          INTEGER  WORD 13
IOCO       INTEGER  WORD  1      I.3          INTEGER  WORD 15
K.1        INTEGER  WORD 18      J.1          INTEGER  WORD 16
K.3        INTEGER  WORD 20      K.2          INTEGER  WORD 19
L.4        INTEGER  WORD  9      K.4          INTEGER  WORD 21
LIST       DOUBLE   WORD  7      L.6          INTEGER  WORD 22
NETM       DOUBLE   WORD 25      S.1          INTEGER  WORD 17
REDUCED.HF DOUBLE   WORD 29      R.1          DOUBLE   WORD 11
HST        DOUBLE   WORD  3      REMAINING.   DOUBLE   WORD 27
                                 TM1          DOUBLE   WORD 23
```

```
LINE  CACI SIMSCRIPT II.5  RELEASE 8F                          09/01/76  PAGE  29

      ROUTINE RATE(M,S,N)
      ''
      '' GENERATES ELEMENT CRACK PROPAGATION RATES REFLECTING VARIATION IN MATERIAL
      ''    PROPERTIES AND LOAD ENVIRONMENT VARIATION
      ''
      '' DISTRIBUTION OF CRACK PROPAGATION RATES IS REPRESENTED AS NORMAL
      ''    M= MEAN PROPAGATION RATE
      ''    S = STANDARD DEVIATION
      ''    N = RANDOM NUMBER STREAM
      ''
      '' METHOD BASED ON APPROXIMATIONS IN C. HASTINGS, APPROXIMATIONS FOR DIGITAL
      ''    COMPUTERS
      ''
      '' IF PROCEDURE YIELDS NEGATIVE RATE, RATE IS SET EQUAL TO MEAN MINUS FOUR
      ''    STANDARD DEVIATIONS
      ''
            DEFINE N AS AN INTEGER VARIABLE
            LET RN = 1.0 - RANDOM.F(N)
            LET RN1 = RN
          IF RN GT 0.5
            LET RN = 1.0 - RN
          REGARDLESS
            LET W = SQRT.F(LOG.E.F(1.0/RN**2))
            LET G1 = 2.515517 + .802853*W + .010328*W**2
            LET G2 = 1.0 + 1.432788*W + .189269*W**2 + .001308*W**3
            LET Z = W - G1/G2
          IF RN1 GT 0.5
            LET Z = -Z
          REGARDLESS
            LET Z = Z * S + M
          IF Z LE 0
            LET Z = M - (4 * S)
          REGARDLESS
            RETURN WITH Z
            END

                LOCAL VARIABLES OF THIS ROUTINE

G1          DOUBLE   WORD 13       G2          DOUBLE   WORD 15
M           DOUBLE   WORD  1       N           INTEGER  WORD  5
RN          DOUBLE   WORD  7       RN1         DOUBLE   WORD  9
S           DOUBLE   WORD  3       W           DOUBLE   WORD 11
Z           DOUBLE   WORD 17
```

```
EVENT 1.STRENGTH.REDUCTION(IDISR) SAVING THE EVENT NOTICE
''
'' REPRESENTS FIRST CRACK INITIATION
''
     DEFINE IDISR AS AN INTEGER VARIABLE
     LET ID = IDISR
     LET TAR = ENTRY.TIME(AIRPLANE(ID)) + USAGE.LIFE
   IF LTHD = "YES"
     FOR I = 1 TO NMAC(LOX)
     DO
    IF ID = TLID(LOX,I)
     LET LIST = 1.0
     SKIP 1 OUTPUT LINE
     PRINT 1 LINE WITH ID, TIME.V-ENTRY.TIME(AIRPLANE(ID)) AS FOLLOWS
       A/C NO. *** EXPERIENCES 1ST CRACK INITIATION AT ****** HOURS
     LEAVE
    ELSE
     LOOP
   ALWAYS
     LET ICRKT = TIME.V - CKREP.TIME(ID)
     LET GICRK = TIME.V - CKREP.TIME(ID)
     LET OICR = OICR + 1
     LET GOICR = GOICR + 1
     LET 1.INT(ID) = "NO"
    IF RANDOM.F(ID) LE 1.PROB
     LET 1.INT(ID) = "YES"
    IF LIST = 1.0
     PRINT 1 LINE AS FOLLOWS
           CRACK INITATES INTERNALLY
    ALWAYS
     LET T = (CRIT.CRK.LGT*PCCL)/(MSR(ID)*CGRI(ID))
    IF PCCL GT 1.0
     LET T = CRIT.CRK.LGT/(MSR(ID)*CGRI(ID)) + (CRIT.CRK.LGT*(PCCL-1.0))/
     (MFR(ID)*CGRI(ID))
    ALWAYS
    IF TIME.V + T LT TAR OR CU.EXISTS(ID) = "NS"
     SCHEDULE A 1.ITF(ID) AT TIME.V + T
     LET AIE(ID) = 1.ITF
     LET IEI(ID) = "YES"
    ALWAYS
   REGARDLESS
     LET 1.CR.EXISTS(ID) = "YES"
   IF INSP.SCH(ID) = "NO"
     CALL INSPECTION.SCHEDULER(I)
   REGARDLESS
''
'' PREDICT TIME TO FAILURE FROM FIRST CRACK INITIATION
''
     LET SU = 2.75
     LET SF = 1.0
     LET XRN(ID) = RANDOM.F(5)
     LET LG = LOG.E.F(XRN(ID))
     LET GR1 = MSR(ID) * CGRI(ID)
```

```
        LET GR2 = MFR(ID) * CGR1(ID)
        LET T1 = CRIT.CRK.LGT / GR1
        LET T2 = T1 + (FSAF.LGT - CRIT.CRK.LGT) / GR2
        LET R1 = ((SU-SF)*GR1) / FSAF.LGT
        LET R2 = ((SU-SF)*GR2) / FSAF.LGT
        LET R3 = GR2 / (LGHT.TO.FAILURE - FSAF.LGT)
  ''
      IF TIME.V + T2 LT TAR OR CO.EXISTS(ID) = "NS"
        SCHEDULE A REACH.FAIL.SAFE.LGT(ID) AT TIME.V + T2
        LET ARFSL(ID) = REACH.FAIL.SAFE.LGT
        LET AIL(ID) = "YES"
      REGARDLESS
        LET K1 = A * EXP.F(SU * B)
        LET K2 = B * R1
        LET TTF = -LOG.E.F((K2/K1)*LG + 1.0)/K2
       IF TTF LE T1
        JUMP AHEAD
       ELSE
        LET S1 = SU - R1 * T1
        LET K4 = B * R2
        LET LGKS = LOG.E.F(A) + (B * S1 + B * R2 * T1)
        LET K8 = (A*EXP.F(B*S1))/K4
        LET K9 = (K1/K2)*(EXP.F(-B*R1*T1)-1.0)
        LET TTF = (LGKS - LOG.E.F(ABS.F(K4)) - LOG.E.F(ABS.F(LG*K8-K9)))/K4
       IF TTF LE T2
        JUMP AHEAD
       ELSE
        LET K10 = B*R3
        LET ARG = B*SF+B*R3*T2
      IF ARG LT -175.0
        LET ARG = -175.0
      ALWAYS
        LET K11 = K10/(A*EXP.F(ARG))
        LET K12 = A*EXP.F(B*SF)/K10
        LET K13 = EXP.F(B*R2*T1 - B*R2*T2) - 1.0
        LET TTF = -LOG.E.F(K11*(LG+K12-(K8*K13)-K9))/K10
      HERE
       IF LIST = 1.0
        PRINT 1 LINE WITH TIME.V+TTF-ENTRY.TIME(AIRPLANE(ID)) AS FOLLOWS
              ELEMENT FAILURE PROJECTED AT ****** FLIGHT HOURS
       ALWAYS
      IF TIME.V + TTF LT TAR OR CO.EXISTS(ID) = "NS"
        SCHEDULE A FAILURE(ID) AT TIME.V + TTF
        LET AF(ID) = FAILURE
        LET FSN(ID) = "YES"
      ALWAYS
        RETURN
        END
```

```
LINE  CACI SIMSCRIPT II.5  RELEASE 6F

                LOCAL VARIABLES OF THIS ROUTINE                                       09/01/76  PAGE  32

 ARG          DOUBLE    WORD 61      GK1          DOUBLE    WORD 29
 GR2          DOUBLE    WORD 31      I.1          INTEGER   WORD 11
 I.2          INTEGER   WORD 12      I.3          INTEGER   WORD 13
 IDISR        INTEGER   WORD  1      J.1          INTEGER   WORD 14
 K.1          INTEGER   WORD 16      K.2          INTEGER   WORD 17
 K.3          INTEGER   WORD 18      K.4          INTEGER   WORD 19
 K1           DOUBLE    WORD 43      K10          DOUBLE    WORD 59
 K11          DOUBLE    WORD 63      K12          DOUBLE    WORD 65
 K13          DOUBLE    WORD 67      K2           DOUBLE    WORD 45
 K4           DOUBLE    WORD 51      K8           DOUBLE    WORD 55
 K9           DOUBLE    WORD 57      L.4          INTEGER   WORD  7
 LG           DOUBLE    WORD 27      LGK5         DOUBLE    WORD 53
 LIST         DOUBLE    WORD  5      N.1          INTEGER   WORD 15
 R.1          DOUBLE    WORD  9      R1           DOUBLE    WORD 37
 R2           DOUBLE    WORD 39      R5           DOUBLE    WORD 41
 SF           DOUBLE    WORD 25      SU           DOUBLE    WORD 23
 S1           DOUBLE    WORD 49      T            DOUBLE    WORD 21
 TAR          DOUBLE    WORD  3      TIF          DOUBLE    WORD 47
 T1           DOUBLE    WORD 33      T2           DOUBLE    WORD 35
```

```
EVENT 2.STRENGTH.REDUCTION(I02SR) SAVING THE EVENT NOTICE
''
'' REPRESENTS SECOND CRACK INITIATION
''
        DEFINE I02SR AS AN INTEGER VARIABLE
        DEFINE X, Y, W AS REAL, 1-DIMENSIONAL ARRAYS
        RESERVE X(*), Y(*), W(*) AS 5
        LET ID = I02SR
        LET TAW = ENTRY.TIME(AIRPLANE(ID)) + USAGE.LIFE
      IF LTWO = "YES"
        FOR I = 1 TO NOAC(LOK)
        DO
       IF ID = TLID(LOK,I)
        LET LIST = 1.0
        SKIP 1 OUTPUT LINE
        PRINT 1 LINE WITH ID, TIME.V-ENTRY.TIME(AIRPLANE(ID)) AS FOLLOWS
          A/C NO. *** EXPERIENCES 2ND CRACK INITIATION AT ****** HOURS
        LEAVE
       ELSE
        LOOP
      ALWAYS
        LET 2.INI(ID) = "NO"
       IF RANDOM.F(10) LE 1.PROB
        LET 2.INI(ID) = "YES"
       IF LIST = 1.0
        PRINT 1 LINE AS FOLLOWS
              CRACK INITIATES INTERNALLY
       ALWAYS
        LET T = (CRIT.CRK.LGT*PCCL)/(MSR(ID)*CGRI(ID))
       IF PCCL GT 1.0
        LET T = CRIT.CRK.LGT/(MSR(ID)*CGRI(ID)) + (CRIT.CRK.LGT*(PCCL-1.0))/
        (MSR(ID)*CGRI(ID))
       ALWAYS
       IF TIME.V + T LT TAW OR CO.EXISTS(ID) = "NO"
        SCHEDULE A 2.ITF(ID) AT TIME.V + T
        LET A2E(ID) = 2.ITF
        LET IF2(ID) = "YES"
       ALWAYS
      REGARDLESS
        LET 2.CR.EXISTS(ID) = "YES"
        LET 1.STRENGTH.REDUCTION = A1SR(ID)
        LET CCL = CRIT.CRK.LGT
        LET TA1 = TIME.A(1.STRENGTH.REDUCTION)
        LET TCL = CCL/(MSR(ID)*CGRI(ID))
''
'' DETERMINE RESIDUAL STRENGTH REDUCTION BECAUSE OF FIRST CRACK
''
      IF CO.EXISTS(ID) = "YES"
        LET CORROSION = AC(ID)
        LET TAC = TIME.A(CORROSION)
       IF TAC LE TA1
        LET CL = (TIME.V-TA1) * MSR(ID) * CGRI(ID)
       IF CL GT CCL
```

```
     LET CL = CCL + ((CL-CCL)/MSR(ID))*MFR(ID)
      JUMP AHEAD
    OTHERWISE
     JUMP AHEAD
   OTHERWISE
     LET DL = (TAC-TA1) * MSR(ID)
   IF CCL LT DL
     LET CL = CCL + ((DL-CCL)/MSR(ID))*MFR(ID) + (TIME.V-TAC)*MFR(ID)*CGRI(ID)
     JUMP AHEAD
   OTHERWISE
     LET CL = DL + (TIME.V-TAC)*MSR(ID)*CGRI(ID)
   IF CCL LT CL
     LET CL = CCL + ((CL-CCL)/MSR(ID))*MFR(ID)
   REGARDLESS
     JUMP AHEAD
  OTHERWISE
     LET CL = (TIME.V-TA1) * MSR(ID)
   IF CL GT CCL
     LET CL = CCL + ((CL-CCL)/MSR(ID))*MFR(ID)
   REGARDLESS
   HERE
     LET SU = 2.75 - (1.75 * CL) / FSAF.LGT
   IF CL GT FSAF.LGT
     LET SU = 1.0 - (CL-FSAF.LGT)/(LGHT.TO.FAILURE-FSAF.LGT)
   REGARDLESS
''
'' PREDICT TIME TO FAILURE FROM SECOND CRACK INITIATION
''
      LET T1 = 0.0
    IF CL LT CCL
      LET T1 = (CCL-CL) / (MSR(ID) * CGRI(ID))
      LET GR1 = 2.0 * MSR(ID) * CGRI(ID)
     IF TAR LE TIME.V + TCL
      LET GR2 = (MSR(ID) + MFR(ID)) * CGRI(ID)
      JUMP AHEAD
     OTHERWISE
      LET X(1) = 0.0
      LET Y(1) = 0.0
      LET X(2) = TCL-T1
      LET Y(2) = CCL + X(2) * MFR(ID) * CGRI(ID) - T1 * MSR(ID) * CGRI(ID)
      LET X(3) = TAR - TIME.V - T1
      LET Y(3) = Y(2) + (2.0 * MFR(ID) * CGRI(ID))
      LET N = 3
      JUMP AHEAD
    REGARDLESS
     IF TAR LE TIME.V + TCL
      LET GR1 = (MSR(ID) + MFR(ID)) * CGRI(ID)
      LET GR2 = GR1
      JUMP AHEAD
     OTHERWISE
      LET X(1) = 0.0
      LET Y(1) = 0.0
      LET X(2) = TCL
```

```
          LET Y(2) = CCL + TCL*MFR(ID)*CGWI(ID)
          LET X(5) = TAU - TIME.V
          LET Y(5) = Y(2) + (X(5)-TCL) * 2.0 * MFR(ID) * CGWI(ID)
          LET N = 5
        HERE
        IF N GE 5
          LET W(1) = 100.0
          FOR I = 2 TO N
          DO
          LET W(I) = 1.0
          LOOP
          FOR I = 1 TO N
          DO
          LET WX = WX + W(I) * X(I)
          LET WXY = WXY + W(I) * X(I) * Y(I)
          LET WY = WY + W(I) * Y(I)
          LET WXS = WXS + W(I) * X(I)**2
          LET SW = SW + W(I)
          LOOP
          LET GR2 = (SW*WXY - WY*WX) / (SW*WXS - WX**2)
        REGARDLESS
          LET T2 = T1 + (FSAF.LGT - (GR1*T1))/GR2
         IF CL GE CCL
          LET GR1 = GR2
          LET T2 = (FSAF.LGT-CL)/GR2
         REGARDLESS
          LET SF = 1.0
          LET R1 = ((SJ-SF)*GR1) / FSAF.LGT
          LET R2 = ((SJ-SF)*GR2) / FSAF.LGT
          LET R3 = GR2 / (LGHT.TO.FAILURE - FSAF.LGT)
        IF CL GE FSAF.LGT
          LET R1 = GR1 / (LGHT.TO.FAILURE - FSAF.LGT)
          LET R2 = GR2 / (LGHT.TO.FAILURE - FSAF.LGT)
          LET R3 = R2
          LET T2 = 0.0
        REGARDLESS
        IF (TIME.V + T2 LT TAU OR CO.EXISTS(ID) = "NS") AND T2 GT 0.0
         IF AIL(ID) = "YES"
          LET REACH.FAIL.SAFE.LGT = ARFSL(ID)
          CANCEL THE REACH.FAIL.SAFE.LGT
          RESCHEDULE THE REACH.FAIL.SAFE.LGT(ID) AT TIME.V + T2
          JUMP AHEAD
         ELSE
          SCHEDULE A REACH.FAIL.SAFE.LGT(ID) AT TIME.V + T2
          LET ARFSL(ID) = REACH.FAIL.SAFE.LGT
          LET AIL(ID) = "YES"
         HERE
        REGARDLESS
          LET LG = LOG.E.F(XGW(ID))
          LET K1 = A * EXP.F(SJ * H)
          LET K2 = H * R1
          LET TTF = -LOG.E.F((K2/K1)*LG + 1.0)/K2
         IF TTF LE T1
```

```
          JUMP AHEAD
         ELSE
          LET S1 = SU - R1 * T1
          LET K4 = B * R2
          LET LGK5 = LOG.E.F(A) + (B * S1 + B * R2 * T1)
          LET K8 = (A*EXP.F(B*S1))/K4
          LET K9 = (K1/K2)*(EXP.F(-B*R1*T1)-1.0)
          LET TTF = (LGK5 - LOG.E.F(ABS.F(K4)) - LOG.E.F(ABS.F(LG+K8-K9)))/K4
         IF TTF LE T2
          JUMP AHEAD
         ELSE
          LET K10 = B*R3
          LET ARG = B*SF+B*R3*T2
        IF ARG LT -175.0
          LET ARG = -175.0
        ALWAYS
          LET K11 = K10/(A*EXP.F(ARG))
          LET K12 = A*EXP.F(B*SF)/K10
          LET K13 = EXP.F(B*R2*T1 - B*R2*T2) - 1.0
          LET TTF = -LOG.E.F(K11*(LG+K12-(K8*K13)-K9))/K10
        HERE
         IF LIST = 1.0
          PRINT 1 LINE WITH TIME.V+TTF-ENTRY.TIME(AIRPLANE(ID)) AS FOLLOWS
                ELEMENT FAILURE PROJECTED AT ****** FLIGHT HOURS
         ALWAYS
        IF TIME.V + TTF LT IAR OR CO.EXISTS(ID) = "NS"
         IF FSM(ID) = "YES"
          LET FAILURE = AF(ID)
          CANCEL THE FAILURE
          RESCHEDULE THE FAILURE(ID) AT TIME.V + TTF
          JUMP AHEAD
         ELSE
          SCHEDULE A FAILURE(ID) AT TIME.V + TTF
          LET AF(ID) = FAILURE
          LET FSM(ID) = "YES"
         HERE
        ALWAYS
          RELEASE X(*), Y(*), W(*)
          RETURN
          END
```

LOCAL VARIABLES OF THIS ROUTINE

```
ARG      DOUBLE    WORD 89      CCL       DOUBLE    WORD 27
CL       DOUBLE    WORD 35      FL        DOUBLE    WORD 37
GR1      DOUBLE    WORD 43      GR2       DOUBLE    WORD 45
I.1      INTEGER   WORD 15      I.2       INTEGER   WORD 16
I.3      INTEGER   WORD 17      ID2SR     INTEGER   WORD  1
J.1      INTEGER   WORD 18      K.1       INTEGER   WORD 20
K.2      INTEGER   WORD 21      K.3       INTEGER   WORD 22
K.4      INTEGER   WORD 23      K1        DOUBLE    WORD 71
K10      DOUBLE    WORD 87      K11       DOUBLE    WORD 91
```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
LINE   CACI SIMSCRIPT II.5  RELEASE 8F                          09/01/76  PAGE  37

  K12         DOUBLE    WORD 93     K13         DOUBLE    WORD 95
  K2          DOUBLE    WORD 73     K4          DOUBLE    WORD 79
  K8          DOUBLE    WORD 83     K9          DOUBLE    WORD 85
  L.Q         INTEGER   WORD 11     L6          DOUBLE    WORD 69
  LOKS        DOUBLE    WORD 81     LIST        DOUBLE    WORD  9
  N           DOUBLE    WORD 47     N.1         INTEGER   WORD 19
  N.1         DOUBLE    WORD 15     N1          DOUBLE    WORD 63
  N2          DOUBLE    WORD 65     R3          DOUBLE    WORD 67
  SF          DOUBLE    WORD 61     SMP         DOUBLE    WORD 57
  SQ          DOUBLE    WORD 39     S1          DOUBLE    WORD 77
  T           DOUBLE    WORD 25     TAC         DOUBLE    WORD 33
  TAN         DOUBLE    WORD  7     TAI         DOUBLE    WORD 29
  TCL         DOUBLE    WORD 31     TIF         DOUBLE    WORD 75
  T1          DOUBLE    WORD 41     T2          DOUBLE    WORD 59
  W           REAL      WORD  5     WX          DOUBLE    WORD 49
  WXS         DOUBLE    WORD 55     WXY         DOUBLE    WORD 51
  WY          DOUBLE    WORD 53     X           REAL      WORD  3
  Y           REAL      WORD  4
```

```
EVENT 3.STRENGTH.REDUCTION(ID3SR) SAVING THE EVENT NOTICE
''
'' REPRESENTS THIRD CRACK INITIATION
''
        DEFINE ID3SR AS AN INTEGER VARIABLE
        DEFINE X, Y, W AS REAL, 1-DIMENSIONAL ARRAYS
        RESERVE X(*), Y(*), W(*) AS 5
        LET ID = ID3SR
        LET TAR = ENTRY.TIME(AIRPLANE(ID)) + USAGE.LIFE
    IF LTHO = "YES"
        FOR I = 1 TO NOAC(LOX)
        DO
     IF ID = TLID(LOX,I)
        LET LIST = 1.0
        SKIP 1 OUTPUT LINE
        PRINT 1 LINE WITH ID, TIME.V-ENTRY.TIME(AIRPLANE(ID)) AS FOLLOWS
          A/C NO. *** EXPERIENCES 3RD CRACK INITIATION AT ****** HOURS
        LEAVE
     ELSE
        LOOP
    ALWAYS
        LET 3.INT(ID) = "NO"
     IF RANDOM.F(10) LE 3.PRO
        LET 3.INT(ID) = "YES"
      IF LIST = 1.0
        PRINT 1 LINE AS FOLLOWS
              CRACK INITATES INTERNALLY
      ALWAYS
       LET T = (CRIT.CRK.LGT*PCCL)/(MSR(ID)*CGRI(ID))
      IF PCCL GT 1.0
       LET T = CRIT.CRK.LGT/(MSR(ID)*CGRI(ID)) + (CRIT.CRK.LGT*(PCCL-1.0))/
       (MFR(ID)*CGRI(ID))
      ALWAYS
      IF TIME.V + T LT TAR OR CO.EXISTS(ID) = "NO"
       SCHEDULE A 3.ITF(ID) AT TIME.V + T
       LET ASE(ID) = 3.ITF
       LET IE3(ID) = "YES"
      ALWAYS
     REGARDLESS
        LET 3.CR.EXISTS(ID) = "YES"
        LET 1.STRENGTH.REDUCTION = A1SR(ID)
        LET 2.STRENGTH.REDUCTION = A2SR(ID)
        LET CCL = CRIT.CRK.LGT
        LET TCL = CCL/(MSR(ID)*CGRI(ID))
        LET TA1 = TIME.A(1.STRENGTH.REDUCTION)
        LET TA2 = TIME.A(2.STRENGTH.REDUCTION)
''
'' DETERMINE RESIDUAL STRENGTH REDUCTION BECAUSE OF FIRST AND SECOND CRACKS
''
    IF CO.EXISTS(ID)= "YES"
        LET CORROSION= AC(ID)
        LET TAC = TIME.A(CORROSION)
     IF TAC LE TA1
```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
        LET 1CL = (TIME.V-TA1) * MSR(ID) * CGRI(ID)
        LET 2CL = (TIME.V-TA2) * MSR(ID) * CGRI(ID)
       IF 1CL GT CCL
        LET 1CL = CCL + ((1CL-CCL)/MSR(ID))*MFR(ID)
       REGARDLESS
       IF 2CL GT CCL
        LET 2CL = CCL + ((2CL-CCL)/MSR(ID))*MFR(ID)
       REGARDLESS
        LET CL = 1CL + 2CL
        JUMP AHEAD
      OTHERWISE
      IF TAC LE TA2
        LET 0L = (TAC-TA1) * MSR(ID)
        LET 1CL = 0L + (TIME.V-TAC)*MSR(ID)*CGRI(ID)
        LET 2CL = (TIME.V-TA2) * MSR(ID) * CGRI(ID)
       IF CCL LT 0L
        LET 2CL = (TIME.V-TA2) * MSR(ID) * CGRI(ID)
       REGARDLESS
       IF 2CL GT CCL
        LET 2CL = CCL + ((2CL-CCL)/MSR(ID))*MFR(ID)
       REGARDLESS
      REGARDLESS
      IF TAC LE TA1
        LET 1CL = (TIME.V-TA1) * MSR(ID) * CGRI(ID)
       IF 1CL GT CCL
        LET 1CL = CCL + ((1CL-CCL)/MSR(ID))*MFR(ID)
       REGARDLESS
      REGARDLESS
        JUMP AHEAD
     OTHERWISE
        LET 1CL = (TIME.V-TA1) * MSR(ID)
      IF 1CL GT CCL
        LET 1CL = CCL + ((1CL-CCL)/MSR(ID))*MFR(ID)
      REGARDLESS
        LET 2CL = (TIME.V-TA2) * MSR(ID)
      IF 2CL GT CCL
        LET 2CL = CCL + ((2CL-CCL)/MSR(ID))*MFR(ID)
      REGARDLESS
      HERE
        LET CL = 1CL + 2CL
        LET SU = 2.75 - (1.75 * CL) / FSAF.LGT
      IF CL GT FSAF.LGT
        LET SU = 1.0 - (CL-FSAF.LGT)/(LGHT.TO.FAILURE-FSAF.LGT)
      REGARDLESS
  ''
  '' PREDICT TIME TO FAILURE FROM THIRD CRACK INITIATION
  ''
        LET T1 = 0.0
      IF 1CL LT CCL
        LET T1 = (CCL-1CL) / (MSR(ID)*CGRI(ID))
        LET GM1 = 3.0 * MSR(ID) * CGRI(ID)
        LET T2 = (CCL-2CL)/ (MSR(ID)*CGRI(ID))
       IF TAW LE TIME.V + T2
```

```
        LET GR2 = (MFR(ID) + 2.0*MSR(ID)) * CGRI(ID)
        JUMP AHEAD
       OTHERWISE
        LET X(1) = 0.0
        LET Y(1) = 0.0
        LET X(2) = T2-T1
        LET Y(2) = X(2) * (MFR(ID) + 2.0*MSR(ID)) * CGRI(ID)
       IF TAR LE TIME.V + TCL
        LET X(3) = TAR - TIME.V - T1
        LET Y(3) = Y(2) + (X(3)-X(2)) * (2.0*MFR(ID) + MSR(ID)) * CGRI(ID)
        LET N = 3
        JUMP AHEAD
       OTHERWISE
        LET X(3) = TCL - T1
        LET XK = X(3) - T2 + T1
        LET Y(3) = (X(3)+XK)*MFR(ID)*CGRI(ID) + (T2-T1+X(3))*MSR(ID)*CGRI(ID)
        LET X(4) = TAR - TIME.V - T1
        LET Y(4) = (X(4)-X(3)) * 3.0 * MFR(ID) * CGRI(ID)
        LET N = 4
        JUMP AHEAD
      OTHERWISE
        LET X(1) = 0.0
        LET Y(1) = 0.0
      IF 2CL LT CCL
        LET T2 = (CCL-2CL)/(MSR(ID)*CGRI(ID))
       IF TAR LE TIME.V + T2
        LET GR2 = (2.0 * MSR(ID) + MFR(ID)) * CGRI(ID)
        LET GR1 = GR2
        JUMP AHEAD
       OTHERWISE
       IF TAR LE TIME.V + TCL
        LET X(2) = T2
        LET Y(2) = T2 * (2.0 * MSR(ID) + MFR(ID)) * CGRI(ID)
        LET X(3) = TAR - TIME.V
        LET Y(3) = Y(2) + (X(3)-X(2)) * (2.0 * MFR(ID) + MSR(ID)) * CGRI(ID)
        LET N = 3
        JUMP AHEAD
       OTHERWISE
        LET X(2) = T2
        LET Y(2) = T2 * (2.0 * MSR(ID) + MFR(ID)) * CGRI(ID)
        LET X(3) = TCL
        LET Y(3) = Y(2) + (TCL-T2) * (2.0 * MFR(ID) + MSR(ID)) * CGRI(ID)
        LET X(4) = TAR - TIME.V
        LET Y(4) = Y(3) + (X(4)-TCL) * (3.0 * MFR(ID)) * CGRI(ID)
        LET N = 4
        JUMP AHEAD
      OTHERWISE
       IF TAR LE TIME.V + TCL
        LET GR2 = (2.0 * MFR(ID) + MSR(ID)) * CGRI(ID)
        LET GR1 = GR2
        JUMP AHEAD
       OTHERWISE
        LET X(2) = TCL
```

```
  LET Y(2) = TCL * (2.0 * MFR(ID) + MSR(ID)) * CGRI(ID)
  LET X(3) = TAR - TIME.V
  LET Y(3) = Y2 + (X(3)-X(2)) * (3.0 * MFR(ID)) * CGRI(ID)
  LET N = 3
HERE
IF N GE 3
  LET W(1) = 100.0
  FOR I = 2 TO N
  DO
  LET W(I) = 1.0
  LOOP
  FOR I = 1 TO N
  DO
  LET WX = WX + W(I) * X(I)
  LET WXY = WXY + W(I) * X(I) * Y(I)
  LET WY = WY + W(I) * Y(I)
  LET WXS = WXS + W(I) * X(I)**2
  LET SMW = SMW + W(I)
  LOOP
  LET GR2 = (SMW*WXY - WY*WX) / (SMW*WXS - WX**2)
REGARDLESS
  LET T2 = T1 + (FSAF.LGT - (GR1*T1))/GR2
 IF TCL GE CCL
  LET GR1 = GR2
  LET T2 = (FSAF.LGT - CL) / GR2
 REGARDLESS
  LET SF = 1.0
  LET R1 = ((SM-SF)*GR1) / FSAF.LGT
  LET R2 = ((SM-SF)*GR2) / FSAF.LGT
  LET R3 = GR2 / (LGHT.TO.FAILURE - FSAF.LGT)
IF CL GE FSAF.LGT
  LET R1 = GR1 / (LGHT.TO.FAILURE - FSAF.LGT)
  LET R2 = GR2 / (LGHT.TO.FAILURE - FSAF.LGT)
  LET R3 = R2
  LET T2 = 0.0
REGARDLESS
IF (TIME.V + T2 LT TAR OR CO.EXISTS(ID) = "NS") AND T2 GT 0.0
 IF AIL(ID) = "YES"
  LET REACH.FAIL.SAFE.LGT = ARFSL(ID)
  CANCEL THE REACH.FAIL.SAFE.LGT
  RESCHEDULE THE REACH.FAIL.SAFE.LGT(ID) AT TIME.V + T2
  JUMP AHEAD
 ELSE
  SCHEDULE A REACH.FAIL.SAFE.LGT(ID) AT TIME.V + T2
  LET ARFSL(ID) = REACH.FAIL.SAFE.LGT
  LET AIL(ID) = "YES"
 HERE
REGARDLESS
  LET K1 = A * EXP.F(SU * M)
  LET K2 = M * R1
  LET LG = LOG.E.F(XRN(ID))
  LET TTF = -LOG.E.F((K2/K1)*LG + 1.0)/K2
 IF TTF LE T1
```

```
       JUMP AHEAD
      ELSE
       LET S1 = S0 - R1 * T1
       LET K4 = B * R2
       LET LGK5 = LOG.E.F(A) + (B * S1 + B * R2 * T1)
       LET K8 = (A*EXP.F(B*S1))/K4
       LET K9 = (K1/K2)*(EXP.F(-B*R1*T1)-1.0)
       LET TTF = (LGK5 - LOG.E.F(ABS.F(K4)) - LOG.E.F(ABS.F(LG+K8-K9)))/K4
      IF TTF LE T2
       JUMP AHEAD
      ELSE
       LET K10 = B*R3
       LET ARG = B*SF+B*R3*T2
     IF ARG LT -175.0
       LET ARG = -175.0
     ALWAYS
       LET K11 = K10/(A*EXP.F(ARG))
       LET K12 = A*EXP.F(B*SF)/K10
       LET K13 = EXP.F(B*R2*T1 - B*R2*T2) - 1.0
       LET TTF = -LOG.E.F(K11*(LG+K12-(K8*K13)-K9))/K10
     HERE
      IF LIST = 1.0
       PRINT 1 LINE WITH TIME.V+TTF-ENTRY.TIME(AIRPLANE(ID)) AS FOLLOWS
             ELEMENT FAILURE PROJECTED AT ****** FLIGHT HOURS
      ALWAYS
     IF TIME.V + TTF LT TAR OR CO.EXISTS(ID) = "NS"
      IF FSH(ID) = "YES"
       LET FAILURE = AF(ID)
       CANCEL THE FAILURE
       RESCHEDULE THE FAILURE(ID) AT TIME.V + TTF
       JUMP AHEAD
      ELSE
       SCHEDULE A FAILURE(ID) AT TIME.V + TTF
       LET AF(ID) = FAILURE
       LET FSH(ID) = "YES"
      HERE
     ALWAYS
       RELEASE X(*), Y(*), Z(*)
       RETURN
       END
```

LOCAL VARIABLES OF THIS ROUTINE

```
ARG         DOUBLE    WORD 99       CCL        DOUBLE    WORD 27
CL          DOUBLE    WORD 41       DL         DOUBLE    WORD 45
GR1         DOUBLE    WORD 49       GR2        DOUBLE    WORD 53
I.1         INTEGER   WORD 15       I.2        INTEGER   WORD 16
I.3         INTEGER   WORD 17       ID.SSS     INTEGER   WORD  1
J.1         INTEGER   WORD 18       K.1        INTEGER   WORD 20
K.2         INTEGER   WORD 21       K.3        INTEGER   WORD 22
K.4         INTEGER   WORD 23       K1         DOUBLE    WORD 79
K10         DOUBLE    WORD 97       K11        DOUBLE    WORD101
```

```
                                                            09/01/76  PAGE  43
LINE  CACI SIMSCRIPT II.5  RELEASE 8F

 K12          DOUBLE    WORD103       K13         DOUBLE    WORD105
 K2           DOUBLE    WORD 81       K4          DOUBLE    WORD 89
 K8           DOUBLE    WORD 93       K9          DOUBLE    WORD 95
 L.4          INTEGER   WORD 11       LG          DOUBLE    WORD 83
 LGKS         DOUBLE    WORD 91       LIST        DOUBLE    WORD  9
 N            DOUBLE    WORD 55       N.1         INTEGER   WORD 19
 R.1          DOUBLE    WORD 13       R1          DOUBLE    WORD 73
 R2           DOUBLE    WORD 75       R5          DOUBLE    WORD 77
 SF           DOUBLE    WORD 71       SMA         DOUBLE    WORD 69
 SU           DOUBLE    WORD 45       S1          DOUBLE    WORD 87
 T            DOUBLE    WORD 25       TAC         DOUBLE    WORD 35
 TAR          DOUBLE    WORD  7       TA1         DOUBLE    WORD 31
 TA2          DOUBLE    WORD 33       TCL         DOUBLE    WORD 29
 TTF          DOUBLE    WORD 85       T1          DOUBLE    WORD 47
 T2           DOUBLE    WORD 51       W           REAL      WORD  5
 XX           DOUBLE    WORD 61       XXS         DOUBLE    WORD 67
 XXY          DOUBLE    WORD 63       XY          DOUBLE    WORD 65
 X            REAL      WORD  3       XK          DOUBLE    WORD 57
 Y            REAL      WORD  4       Y2          DOUBLE    WORD 59
 ZCL          DOUBLE    WORD 37       ZCL         DOUBLE    WORD 39
```

```
LINE   CACI SIMSCRIPT II.5  RELEASE 8F                         09/01/76  PAGE  46

     EVENT LITE(IDIE)
           DEFINE IDIE AS AN INTEGER VARIABLE
           LET ID = IDIE
           LET I.INT(ID) = "EQ"
           LET IFI(ID) = "NO"
        IF (TMO = "YES"
           FOR I = 1 TO NJAC(LDX)
           DO
        IF ID = TLID(LDX,I)
          SKIP 1 OUTPUT LINE
          PRINT 1 DOUBLE LINE WITH ID, CRIT.UNK.LGT*PCCL, TIME.V-
          ENTRY.TI-F(AIRPLANE(ID)) AS FOLLOWS
             AC      * *** HAS INTERNAL FIRST CRACK BECOME EXTERNAL AT **.** INCHES AND ****** FLIGHT HOURS
          LEAVE
        ELSE
          LOOP
       ALWAYS
          RETURN
          END


              LOCAL VARIABLES OF THIS ROUTINE

I.1        INTEGER   WORD  7
I.3        INTEGER   WORD  8       I.2
J.1        INTEGER   WORD 10       IDIE      INTEGER   WORD   4
K.2        INTEGER   WORD 13       K.1       INTEGER   WORD   1
K.4        INTEGER   WORD 15       K.5       INTEGER   WORD  12
N.1        INTEGER   WORD 11       L.4       INTEGER   WORD  14
                                   M.1       INTEGER   WORD   5
                                             DOUBLE    WORD   5
```

```
LINE  CACI SIMSCRIPT II.5  RELEASE 8F                                      09/01/76  PAGE  45

  1  EVENT 2.ITF(ID26)
  2        DEFINE ID26 AS AN INTEGER VARIABLE
  3        LET ID = ID26
  4        LET 2.INT(ID) = "NO"
  5        LET IN2(ID) = "NO"
  6     IF LINK = "YES"
  7        FOR I = 1 TO NSUC(LOX)
  8        DO
  9      IF ID = ILINK(LOX,I)
 10        SKIP 1 OUTPUT LINE
 11        PRINT 1 DOUBLE LINE WITH ID, CHIT.LOW.LLT.HEIL, TIME.V=
 12        ENTRY.TIME(AIRPLANE(ID)) AS FOLLOWS
             A/C NO. *** HAS INTERNAL SECOND CHANCE HIT ** EXTERNAL AT **.** INCHES AND ****** FLIGHT ~ ~~
 13        LEAVE
 14      ELSE
 15        LOOP
 16     ALWAYS
 17        RETURN
 18        END


                 LOCAL VARIABLES OF THIS ROUTINE

  I.1          INTEGER  WORD  7       I.2          INTEGER  WORD  8
  I.3          INTEGER  WORD  9       ID26         INTEGER  WORD  1
  J.1          INTEGER  WORD 10       K.1          INTEGER  WORD 12
  K.2          INTEGER  WORD 13       K.3          INTEGER  WORD 14
  K.4          INTEGER  WORD 15       L.3          INTEGER  WORD  3
  N.1          INTEGER  WORD 11       R.1          DOUBLE   WORD  5
```

```
LINE  CACI SIMSCRIPT II.5  RELEASE 8F                                          09/01/76  PAGE  46

   1  EVENT 3.ITE(ID3E)
   2       DEFINE ID3E AS AN INTEGER VARIABLE
   3       LET ID = ID3E
   4       LET 3.INT(ID) = "NO"
   5       LET IE3(ID) = "VO"
   6     IF LTHO = "YES"
   7       FOR I = 1 TO NOAC(LDX)
   8       DO
   9      IF ID = TLID(LDX,I)
  10       SKIP 1 OUTPUT LINE
  11       PRINT 1 DOUBLE LINE WITH ID, CRIT.CRK.LGT*ACCL, TIME.V-
  12       ENTRY.TIME(AIRPLANE(ID)) AS FOLLOWS
             A/C NO. *** HAS INTERNAL THIRD CRACK BECOME EXTERNAL AT **.** INCHES AND ****** FLIGHT HOURS
  13       LEAVE
  14     ELSE
  15       LOOP
  16    ALWAYS
  17       RETURN
  18       END


              LOCAL VARIABLES OF THIS ROUTINE

 I.1         INTEGER  WORD  7      I.2         INTEGER  WORD  8
 I.3         INTEGER  WORD  9      ID3E        INTEGER  WORD  1
 J.1         INTEGER  WORD 10      K.1         INTEGER  WORD 12
 K.2         INTEGER  WORD 13      K.3         INTEGER  WORD 14
 K.4         INTEGER  WORD 15      L.4         INTEGER  WORD  3
 N.1         INTEGER  WORD 11      K.1         DOUBLE   WORD  5
```

```
LINE  CACI SIMSCRIPT II.5  RELEASE 8F                              09/01/76  PAGE  47

      ROUTINE INSPECTION.SCHEDULER(ID)
      ''
      '' SCHEDULES INSPECTIONS BELOW OVERHAUL LEVEL ON A GIVEN AIRCRAFT SUCH THAT THE
      ''     AIRCRAFT IS NOT INSPECTED PRIOR TO CRACK REACHING MINIMUM DETECTABLE LGT
      ''
             DEFINE N AS AN INTEGER VARIABLE
             LET M1 = MSR(ID) * CSMI(ID)
             LET C1 = C.GROWTH.RATE * CRRF
          IF EXT.INSP.LEVEL LE 1
             LET TML = TIME.V + .551/M1
            IF N = 2
             LET TML = TIME.V + .996/C1
            ALWAYS
             LET S.INSP.AT = ENTRY.TIME(AIRPLANE(ID)) + IAHCD(1)
             * TRUNC.F((TML-ENTRY.TIME(AIRPLANE(ID)))/IAHCD(1) + 1.0)
             SCHEDULE AN A.LEVEL.INSPECTION(ID) AT S.INSP.AT
             LET AAL(ID) = A.LEVEL.INSPECTION
          REGARDLESS
      ''
          IF EXT.INSP.LEVEL LE 2
             LET TML = TIME.V + .410/M1
            IF N = 2
             LET TML = TIME.V + .655/C1
            ALWAYS
             LET S.INSP.AT = ENTRY.TIME(AIRPLANE(ID)) + IAHCD(2)
             * TRUNC.F((TML-ENTRY.TIME(AIRPLANE(ID)))/IAHCD(2) + 1.0)
             SCHEDULE A B.LEVEL.INSPECTION(ID) AT S.INSP.AT
             LET ABL(ID) = B.LEVEL.INSPECTION
          REGARDLESS
      ''
          IF EXT.INSP.LEVEL LE 3
             LET TML = TIME.V + .266/M1
            IF N = 2
             LET TML = TIME.V + .563/C1
            ALWAYS
             LET S.INSP.AT = ENTRY.TIME(AIRPLANE(ID)) + C.INTERVAL(ID)
             * TRUNC.F((TML-ENTRY.TIME(AIRPLANE(ID)))/C.INTERVAL(ID) + 1.0)
             SCHEDULE A C.LEVEL.INSPECTION(ID) AT S.INSP.AT
             LET ACL(ID) = C.LEVEL.INSPECTION
          REGARDLESS
      ''
             LET INSP.SCH(ID) = "YES"
             RETURN
             END
```

```
LINE  CACI SIMSCRIPT II.5  RELEASE 8F

              LOCAL VARIABLES OF THIS ROUTINE                         09/01/76  PAGE  48
C1
N           DOUBLE   WORD  5
TML         INTEGER  WORD  1      K1          DOUBLE   WORD  3
            DOUBLE   WORD  7      S.INSP.AT   DOUBLE   WORD  9
```

```
LINE  CACI SIMSCRIPT II.5  RELEASE 8F                        09/01/76  PAGE  49

   1  EVENT A.LEVEL.INSPECTION(IDA)
   2        DEFINE IDA AS AN INTEGER VARIABLE
   3        LET ID = IDA
   4        LET FIXIT.COST = A.REPAIR.COST
   5        LET RSW = 1.0
   6   ''
   7        CALL EXAMINE(1.,.275,.512,.996,.275,.512,.551) YIELDING FLOOD
   8   ''
   9        SCHEDULE A A.LEVEL.INSPECTION(ID) AT TIME.V + TAHCO(1)
  10        LET LAC(ID) = A.LEVEL.INSPECTION
  11        RETURN
  12        END

          LOCAL VARIABLES OF THIS ROUTINE

 FLOOD     DOUBLE   SUBR  3      IDA        INTEGER  SUBR  1
```

```
LINE  CACI SIMSCRIPT II.5  RELEASE 8F                                    09/01/76  PAGE  50

   1  EVENT B.LEVEL.INSPECTION(IDB)
   2        DEFINE IDB AS AN INTEGER VARIABLE
   3        LET ID = IDB
   4  ''
   5       IF EXT.INSP.LEVEL LE 1
   6        LET A.LEVEL.INSPECTION = AAL(ID)
   7        CANCEL THE A.LEVEL.INSPECTION
   8        RESCHEDULE THE A.LEVEL.INSPECTION(ID) AT TIME.V + IABCD(1)
   9        LET AAL(ID) = A.LEVEL.INSPECTION
  10       ALWAYS
  11        LET FIXIT.COST = B.REPAIR.COST
  12        LET BSMP = 1.0
  13  ''
  14        CALL EXAMINE(2.,.462,.513,.635,.462,.515,.410) YIELDING FOUND
  15  ''
  16        SCHEDULE A B.LEVEL.INSPECTION(ID) AT TIME.V + IABCD(2)
  17        LET BAL(ID) = B.LEVEL.INSPECTION
  18        RETURN
  19        END

             LOCAL VARIABLES OF THIS ROUTINE

 FOUND       DOUBLE   WORD  5       IDB          INTEGER  WORD  1
```

```
LINE  CACI SIMSCRIPT II.5  RELEASE 8F                          09/01/76  PAGE  51

   1  EVENT C.LEVEL.INSPECTION(IOC)
   2        DEFINE IOC AS AN INTEGER VARIABLE
   3        LET ID = IOC
   4  ''
   5       IF EXT.INSP.LEVEL EQ 2
   6         LET B.LEVEL.INSPECTION = BBL(ID)
   7         CANCEL THE B.LEVEL.INSPECTION
   8         RESCHEDULE THE B.LEVEL.INSPECTION(ID) AT TIME.V + IAMC(ID,2)
   9  ''
  10        IF EXT.INSP.LEVEL EQ 1
  11         LET A.LEVEL.INSPECTION = AAL(ID)
  12         CANCEL THE A.LEVEL.INSPECTION
  13         RESCHEDULE THE A.LEVEL.INSPECTION(ID) AT TIME.V + IAMC(ID,1)
  14         LET AAL(ID) = A.LEVEL.INSPECTION
  15        ALWAYS
  16       ALWAYS
  17         LET FIXIT.COST = C.REPAIR.COST
  18         LET RANK = 1.0
  19  ''
  20         CALL EXAMINE(5,.005,.006,.5E3,.003,.666,.200) YIELDING FOUND
  21  ''
  22       IF FOUND = 1
  23         FOR I = 1 TO [illegible]
  24         DO
  25        IF MI.TIME.ACRFT(I) = I
  26         LET MWDO(I) = TIME.V
  27         CYCLE
  28        ELSE
  29         LOOP
  30       REGARDLESS
  31         SCHEDULE A C.LEVEL.INSPECTION(ID) AT TIME.V + C.INTERVAL(ID)
  32         LET ACL(ID) = C.LEVEL.INSPECTION
  33         RETURN
  34         END

              LOCAL VARIABLES OF THIS ROUTINE

FOUND       [illegible]   [illegible] 5       INT          [illegible]   1
```

```
EVENT D.LEVEL.INSPECTION(IDD)
      DEFINE IDD AS AN INTEGER VARIABLE
      DEFINE I AS AN INTEGER VARIABLE
      LET ID = IDD
   IF LT[illegible] = "YES"
      FOR I = 1 TO NDAC(LDK)
      DO
    IF ID = TLID(LDK,I)
      SKIP 1 OUTPUT LINE
      PRINT 1 LINE WITH ID, TIME.V-ENTRY.TIME(AIRPLANE(ID)) AS FOLLOWS
        D-LEVEL INSPECTION [illegible] A/C [illegible] , *** AT ****** HOURS
      LEAVE
    ELSE
      LOOP
   ALWAYS
  IF INSP.SCH(ID) = "YES"
   IF EXT.INSP.LEVEL LE 3
      LET C.LEVEL.INSPECTION = ACL(ID)
      CANCEL THE C.LEVEL.INSPECTION
      RESCHEDULE THE C.LEVEL.INSPECTION(ID) AT TIME.V + C.INTERVAL(ID)
''
    IF EXT.INSP.LEVEL LE 2
      LET B.LEVEL.INSPECTION = BCL(ID)
      CANCEL THE B.LEVEL.INSPECTION
      RESCHEDULE THE B.LEVEL.INSPECTION(ID) AT TIME.V + IANCH(2)
''
     IF EXT.INSP.LEVEL LE 1
      LET A.LEVEL.INSPECTION = AAL(ID)
      CANCEL THE A.LEVEL.INSPECTION
      RESCHEDULE THE A.LEVEL.INSPECTION(ID) AT TIME.V + IANCH(1)
     ALWAYS
    ALWAYS
   ALWAYS
  REGARDLESS
      LET FINIT.COST = [illegible].COST
      LET [illegible] = SAVPIT [illegible]
''
      CALL EXAMINE([illegible],.993,.725,.351,.990,.725,.144) YIELDING [illegible]
''
      LET I = MI.TIME.ACRFT(ID)
   IF I GE 10 AND TIME.V - [illegible].TIME(AIRPLANE(ID)) GE ANC(4)
      FOR I = 1 TO 10
      DO
    IF MI.TIME.ACRFT(I) = ID
     IF FO[illegible] = 1
      LET [illegible](I) = TIME.V
     ALWAYS
      FOR J = 1 TO 10
      DO
     IF TIME.V - [illegible](I) LT ANCH(4)
      GO TO NUCHK
     ELSE
      LOOP
```

```
        LET AMCD(3) = AMCD(3) + FREQ.DECREASE
        LET AMCD(4) = AMCD(4) + FREQ.DECREASE
       IF LIMO = "YES"
        SKIP 1 OUTPUT LINE
        PRINT 3 LINES WITH AMCD(3), AMCD(4) AS FOLLOWS
           I.SPECTI . INTERVAL INCREASE IMPLEMENTED
               C-LEVEL INTERVAL NOW ***** HOURS
               D-LEVEL INTERVAL NOW ***** HOURS
       ALWAYS
        LET NICHG = NICHG +1
       IF NICHG GT 50
        JUMP AHEAD
       ELSE
        LET SC(NICHG) = AMCD(3)
        LET SD(NICHG) = AMCD(4)
       HERE
        LET CBASE = AMCD(3)
        LET DBASE = AMCD(4)
        FOR EVERY AIRCRAFT IN ACTIVE.FLEET
        DO
        LET C.INTERVAL(TAIL.ID) = AMCD(3)
        LET D.INTERVAL(TAIL.ID) = AMCD(4)
        LOOP
        FOR J = 1 TO 10
        DO
        LET MROD(J) = TIME.V
        LOOP
'NOCHG' LEAVE
     ELSE
        LOOP
    ALWAYS
''
  IF TMOD.PENDING(ID) = "YES" OR SMOD.PENDING(ID) = "YES"
        SCHEDULE A D.LEVEL.INSPECTION(ID) AT TIME.V + D.INTERVAL(ID)
        LET ADL(ID) = D.LEVEL.INSPECTION
    IF TMOD.PENDING(ID) = "YES"
        CALL INSTALL.MODIFICATION
        LET TMOD.PENDING(ID) = "NO"
     IF TIME.V LT ENTRY.TIME(AIRPLANE(ID)) + C7 * 144FL
        LET I.INSPECTION.INCREASE = ATII(ID)
        CANCEL THE I.INSPECTION.INCREASE
        DESTROY THE I.INSPECTION.INCREASE
     REGARDLESS
        RETURN
    OTHERWISE
        CALL INSTALL.MODIFICATION
        LET SMOD.PENDING(ID) = "NO"
        RETURN
   OTHERWISE
''
     IF FOUND GT 0
        SCHEDULE A REPAIR(ID) NOW
     REGARDLESS
```

```
LINE  CACI SIMSCRIPT II.5  RELEASE 8                              09/01/76  PAGE 58

 103        SCHEDULE A D.LEVEL.INSPECTION(ID) AT TIME.V + D.INTERVAL(ID)
 104        LET SOL(ID) = D.LEVEL.INSPECTION
 105        RETURN
 106        END

                 LOCAL VARIABLES OF THIS ROUTINE:

FIGURE      DOUBLE    WORD 17       I.1         INTEGER   WORD  7
I.2         INTEGER   WORD  8       I.3         INTEGER   WORD  9
I.99        INTEGER   WORD  1       J           INTEGER   WORD  3
J.1         INTEGER   WORD 10       K.1         INTEGER   WORD 12
K.2         INTEGER   WORD 14       K.3         INTEGER   WORD 14
K.4         INTEGER   WORD 15       L.12        INTEGER   WORD 19
L.17        INTEGER   WORD 20       L.4         INTEGER   WORD  6
M.1         INTEGER   WORD 11       M.1         DOUBLE    WORD  5
```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
ROUTINE FRAMING(N,XA,YA,ZA,XL,YL,ZL) YIELDING FOUND
        DEFINE N AS AN INTEGER VARIABLE
        DEFINE Z AS AN ALPHA VARIABLE
    IF ITMS = "YES"
        FOR I = 1 TO NOAC(L.A)
        DO
     IF ID = ILIN(LOC,I)
        LET LIST = 1.0
        LEAVE
     ELSE
        LOOP
    ALWAYS
        LET H1 = NH(ID)
        LET H2 = FH(ID)
        LET CCL = CRIT.CRK.LGT
   IF C.EXISTS(ID) = "YES"
    IF C.INIT(ID) = "NO" OR C.INIT(ID) = "YES" AND ITMS LE 5
        LET CORROSION = AC(ID)
        LET TAC = TIME.A(CORROSION)
        LET AREA = C.GROWTH.RATE * CONF * (TIME.V - TIME.A(CORROSION))
     IF RANDOM.F(1) LE PROB(XA,YA,ZA,AREA)
        LET DIC(N) = DIC(N) + 1
        LET GDIC(N) = GDIC(N) + 1
        IF N = 1        LET ACA = AREA
                        LET GACA = AREA
                        LET Z = "A"
                        JUMP AHEAD            OTHERWISE
        IF N = 2        LET BCA = AREA
                        LET GBCA = AREA
                        LET Z = "B"
                        JUMP AHEAD            OTHERWISE
        IF N = 3        LET CCA = AREA
                        LET GCCA = AREA
                        LET Z = "C"
                        JUMP AHEAD            OTHERWISE
                        LET DCA = AREA
                        LET GDCA = AREA
                        LET Z = "D"
                                              HERE
      IF LIST = 1.0
       SKIP 1 OUTPUT LINE
       PRINT 1 DOUBLE LINE WITH AREA, Z, ID, TIME.V-ENTRY.TIME(AIRPLANE(ID)) AS
       FOLLOWS
         CORROSION OF AREA **.** SQ. INCHES DETECTED DURING *-LEVEL INSPECTION OF A/C ***, *** AT ****** FLIGHT HOURS
      ALWAYS
       LET FOUND = 2
     REGARDLESS
    REGARDLESS
   REGARDLESS
  ''
  IF I.CR.EXISTS(ID) = "YES"
     IF I.INIT(ID) = "NO" OR I.INIT(ID) = "YES" AND ITMS LE
       LET I.STRENGTH.REDUCTION = AISR(ID)
```

```
         LET TA1 = TIME.AC1.STRENGTH.REDUCTION)
         LET CL = (TIME.V-TA1) * M1 * CGRI(ID)
        IF TAC GT TA1
         LET CL = (TAC-TA1)*M1 + (TIME.V-TAC)*M1*CGRI(ID)
        REGARDLESS
        IF CL GT CCL
         LET CL = CCL + ((CL-CCL)/M1)**2
        REGARDLESS
  ''
        IF RANDOM.F(1) LE POOD(XL,YL,ZL,CL)
         LET DICK(N) = DICK(N) + 1
         LET GDICK(N) = GDICK(N) + 1
         IF N = 1        LET ACHAL = CL
                         LET GACHK = CL
                         LET Z = "A"
                         JUMP AHEAD          OTHERWISE
         IF N = 2        LET BCHAL = CL
                         LET GBCHK = CL
                         LET Z = "B"
                         JUMP AHEAD          OTHERWISE
         IF N = 3        LET CCHAL = CL
                         LET GCCHK = CL
                         LET Z = "C"
                         JUMP AHEAD          OTHERWISE
                         LET DCHAL = CL
                         LET GDCHK = CL
                         LET Z = "D"
                                           HERE
        IF LIST = 1.0
         SKIP 1 OUTPUT LINE
         PRINT 1 DOUBLE LINE WITH CL, Z, ID, TIME.V-ENTRY.TIME(AIRPLANE(ID)) AS
         FOLLOWS
           CRACK OF LENGTH **.** INCHES DETECTED DURING *-LEVEL INSPECTION OF A/C NO. *** AT ****** FLIGHT HOURS
        ALWAYS
         LET FOUND = 1
        REGARDLESS
       ALWAYS
  ''
     IF 2.CM.EXISTS(ID) = "YES"
       IF 2.INT(ID) = "NO" OR 2.INT(ID) = "YES" AND ITEML LE 5
         LET 2.STRENGTH.REDUCTION = A2SR(ID)
         LET TA2 = TIME.A(2.STRENGTH.REDUCTION)
         LET CL = (TIME.V-TA2) * M1 * CGRI(ID)
        IF TAC GT TA2
         LET CL = (TAC-TA2)*M1 + (TIME.V-TAC)*M1*CGRI(ID)
        REGARDLESS
        IF CL GT CCL
         LET CL = CCL + ((CL-CCL)/M1)**2
        REGARDLESS
  ''
        IF RANDOM.F(1) LE POOD(XL,YL,ZL,CL)
         LET DICK(N) = DICK(N) + 1
         LET GDICK(N) = GDICK(N) + 1
```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
          IF N = 1        LET ACHRL = CL
                          LET SACHR = CL
                          LET Z = "A"
                          JUMP AHEAD               OTHERWISE
          IF N = 2        LET HCRRL = CL
                          LET GHCHR = CL
                          LET Z = "H"
                          JUMP AHEAD               OTHERWISE
          IF N = 3        LET CCHRL = CL
                          LET GCCHR = CL
                          LET Z = "C"
                          JUMP AHEAD               OTHERWISE
                          LET TCHRL = CL
                          LET GTCHR = CL
                          LET Z = "D"
                                                   HERE
        IF LIST = YES
         SKIP 1 OUTPUT LINE
         PRINT 1 DOUBLE LINE WITH CL, Z, ID, TIME.V-ENTRY.TIME(AIRPLANE(ID)) AS
         FOLLOWS
           CRACK OF LENGTH **.** INCHES DETECTED DURING *-LEVEL INSPECTION OF A/C NO. *** AT ****** FLIGHT HOURS
        ALWAYS
         LET FOUND = 1
        REGARDLESS
       ALWAYS
 ''
      IF $.LW.EXISTS(ID) = "YES"
       IF $.INT(ID) = "NO" OR $.INT(ID) = "YES" AND TIMEL LE 5
         LET $.STRENGTH.REDUCTION = ASS(ID)
         LET TAS = TIME.A($.STRENGTH.REDUCTION)
         LET CL = (TIME.V-TAS) * M1 * CRI(ID)
        IF TAC GT TAS
         LET CL = (TAC-TAS)*M1 + (TIME.V-TAC)*M1*CRI(ID)
        REGARDLESS
        IF CL GT CCL
         LET CL = CCL + ((CL-CCL)/M1)*M2
        REGARDLESS
 ''
        IF RANDOM.F(1) LE POD(XL,YL,ZL,CL)
         LET DIC(N) = DIC(N) + 1
         LET QUICK(N) = QUICK(N) + 1
          IF N = 1        LET ACHRL = CL
                          LET SACHR = CL
                          LET Z = "A"
                          JUMP AHEAD               OTHERWISE
          IF N = 2        LET HCHRL = CL
                          LET GHCHR = CL
                          LET Z = "H"
                          JUMP AHEAD               OTHERWISE
          IF N = 3        LET CCHRL = CL
                          LET GCCHR = CL
                          LET Z = "C"
                          JUMP AHEAD               OTHERWISE
```

```
LINE  CACI SIMSCRIPT II.5  RELEASE 8F                                     09/01/76  PAGE  58

 157
 158                     LET RCHAL = CL
 159                     LET CHECK = CL
 160                     LET Z = "D"
 161         IF LIST = 1.0                        HERE
 162          SKIP 1 OUTPUT LINE
 163          PRINT 1 DOUBLE LINE WITH CL, Z, ID, TIME.V-ENTRY.TIME(AIRPLANE(ID)) AS
 164          FOLLOWS
               CHECK OF LENGTH **.** INCHES DETECTED DURING *-LEVEL INSPECTION OF A/C NO. *** AT ****** FLIGHT HOURS
 165         ALWAYS
 166          LET FOUND = 1
 167         REGARDLESS
 168        ALWAYS
 169       ALWAYS
 170      ALWAYS
 171     ALWAYS
 172   ''
 173        IF N = 4
 174   ''                         RETURN                  OTHERWISE
 175        IF FOUND GT 0
 176          SCHEDULE A REPAIR(ID) NOW
 177        REGARDLESS
 178          RETURN
 179          END


            LOCAL VARIABLES OF THIS ROUTINE

AREA        DOUBLE     WORD 20        CCL       DOUBLE     WORD 25
CL          DOUBLE     WORD 47        FOUND     DOUBLE     WORD 15
I.1         INTEGER    WORD 35        I.2       INTEGER    WORD 36
I.3         INTEGER    WORD 37        J.1       INTEGER    WORD 38
K.1         INTEGER    WORD 40        K.2       INTEGER    WORD 41
K.3         INTEGER    WORD 42        K.4       INTEGER    WORD 43
L.10        INTEGER    WORD 57        L.2       INTEGER    WORD 51
L.6         INTEGER    WORD 49        L.8       INTEGER    WORD 53
LIST        DOUBLE     WORD 19        M1        DOUBLE     WORD 21
M2          DOUBLE     WORD 23        N         INTEGER    WORD  1
N.1         INTEGER    WORD 39        R.1       DOUBLE     WORD 33
TAC         DOUBLE     WORD 27        TA1       DOUBLE     WORD 45
TA2         DOUBLE     WORD 51        TA5       DOUBLE     WORD 55
XA          DOUBLE     WORD  5        XL        DOUBLE     WORD  9
YA          DOUBLE     WORD  5        YL        DOUBLE     WORD 11
Z           ALPHA      WORD 17        ZA        DOUBLE     WORD  7
ZL          DOUBLE     WORD 14
```

```
                                                                    09/01/76  PAGE  59
LINE  CACI SIMSCRIPT II.5  RELEASE 9F

  1   ROUTINE PROB(X,Y,Z,L)
  2   ''
  3   '' COMPUTES PROBABILITY OF DETECTING A SPECIFIC DEFECT OF A GIVEN SIZE
  4   ''     X = MAXIMUM PROBABILITY OF DETECTION AT A GIVEN INSPECTION LEVEL
  5   ''     Y = EXPONENTIAL PARAMETER - CONSTANT FOR EACH LEVEL OF INSPECTION
  6   ''     Z = MINIMUM SIZE OF DEFECT DETECTABLE AT EACH INSPECTION LEVEL
  7   ''     L = SIZE OF DEFECT BEING TESTED:
  8   ''              LENGTH FOR CRACK DEFECTS
  9   ''              AREA FOR CORROSION DEFECTS
 10   ''
 11         RETURN WITH X * (1.0 - EXP.F(-Y * (L - Z))) * ASEP
 12         END

            LOCAL VARIABLES OF THIS ROUTINE

                                                        DOUBLE    WORD  1
 L          DOUBLE    WORD  7        X                  DOUBLE    WORD  5
 Y          DOUBLE    WORD  5        Z
```

```
LINE  CACI SIMSCRIPT II.5  RELEASE 8F

   ROUTINE CANCEL.SCHEDULED.INSPECTIONS                                   09/01/76  PAGE  60
   ''
   '' CANCELS ALL SCHEDULED INSPECTIONS BELOW OVERHAUL LEVEL ON A GIVEN AIRCRAFT
   ''
      IF EXT.INSP.LEVEL LE 3
         LET C.LEVEL.INSPECTION = ACL(ID)
         CANCEL THE C.LEVEL.INSPECTION
         DESTROY THE C.LEVEL.INSPECTION
   ''
       IF EXT.INSP.LEVEL LE 2
        LET B.LEVEL.INSPECTION = ABL(ID)
        CANCEL THE B.LEVEL.INSPECTION
        DESTROY THE B.LEVEL.INSPECTION
   ''
        IF EXT.INSP.LEVEL LE 1
        LET A.LEVEL.INSPECTION = AAL(ID)
        CANCEL THE A.LEVEL.INSPECTION
        DESTROY THE A.LEVEL.INSPECTION
       ALWAYS
      ALWAYS
     ALWAYS
   ''
        LET INSP.SCH(ID) = "NO"
        RETURN
        END
```

```
LINE  CACI SIMSCRIPT II.5  RELEASE 8F                         09/01/76  PAGE  01

     EVENT REACH.FAIL.SAFE.LGT(IORES)
           DEFINE IORES AS AN INTEGER VARIABLE
           LET ID = IORES
        IF LTHD = "YES"
           FOR I = 1 TO NNAC(LDX)
           DO
         IF ID = TLIN(LDX,I)
           SKIP 1 OUTPUT LINE
           PRINT 1 LINE WITH ID, TIME.V-ENTRY.TIME(AIRPLANE(ID)) AS FOLLOWS
             A/C NO. *** REACHES FAIL-SAFE STRENGTH AT ****** FLIGHT HOURS
           LEAVE
         ELSE
           LOOP
        ALWAYS
          IF NRES LT 20
           LET NRES = NRES + 1
           LET APID(NRES) = ID
           LET STIM(NRES) = TIME.V - ENTRY.TIME(AIRPLANE(ID))
          REGARDLESS
     ''
          IF SNRES LT 200
           LET SNRES = SNRES + 1
           LET SAPID(SNRES) = ID
           LET SSTIM(SNRES) = TIME.V - ENTRY.TIME(AIRPLANE(ID))
           LET SFLNB(SNRES) = ELEVE.I(L)
          REGARDLESS
           LET AIL(ID) = "NO"
           RETURN
           END

              LOCAL VARIABLES OF THIS ROUTINE

  I.1        INTEGER   WORD  7        I.2        INTEGER   WORD  6
  I.3        INTEGER   WORD  9        IORES      INTEGER   WORD  1
  J.1        INTEGER   WORD 10        K.1        INTEGER   WORD 12
  K.2        INTEGER   WORD 13        K.3        INTEGER   WORD 14
  K.4        INTEGER   WORD 15        L.4        INTEGER   WORD  3
  N.1        INTEGER   WORD 11        R.1        DOUBLE    WORD  5
```

```
EVENT FAILURE(IDFA)
      DEFINE IDFA AS AN INTEGER VARIABLE
      DEFINE HOLD AS AN INTEGER VARIABLE
      LET ID = IDFA
   IF LTHO = "YES"
      FOR I = 1 TO NOAC(LDX)
      DO
    IF ID = TLID(LDX,I)
      SKIP 1 OUTPUT LINE
      PRINT 1 LINE WITH ID, TIME.V-ENTRY.TIME(AIRPLANE(ID)) AS FOLLOWS
        A/C NO. *** EXPERINCES ELEMENT FAILURE AT ****** FLIGHT HOURS
      LET 1.STRENGTH.REDUCTION = A1SR(ID)
      LET TA1 = TIME.A(1.STRENGTH.REDUCTION)
      LET CCL = CRIT.CRK.LGT
     IF CO.EXISTS(ID) = "YES"
      LET CORROSION = AC(ID)
      LET TAC = TIME.A(CORROSION)
     ALWAYS
      LET 1CL = (TIME.V-TA1)*MSR(ID)*CGRI(ID)
     IF TAC GT TA1
      LET 1CL = (TAC-TA1)*MSR(ID) + (TIME.V-TAC)*MSR(ID)*CGRI(ID)
     ALWAYS
     IF 1CL GT CCL
      LET 1CL = CCL + (1CL-CCL)*(MFR(ID)/MSR(ID))
     ALWAYS
   IF 2.CR.EXISTS(ID) = "YES"
      LET 2.STRENGTH.REDUCTION = A2SR(ID)
      LET TA2 = TIME.A(2.STRENGTH.REDUCTION)
      LET 2CL = (TIME.V-TA2)*MSR(ID)*CGRI(ID)
     IF TAC GT TA2
      LET 2CL = (TAC-TA2)*MSR(ID) + (TIME.V-TAC)*MSR(ID)*CGRI(ID)
     ALWAYS
     IF 2CL GT CCL
      LET 2CL = CCL + (2CL-CCL)*(MFR(ID)/MSR(ID))
     ALWAYS
    IF 3.CR.EXISTS(ID) = "YES"
      LET 3.STRENGTH.REDUCTION = A3SR(ID)
      LET TA3 = TIME.A(3.STRENGTH.REDUCTION)
      LET 3CL = (TIME.V-TA3)*MSR(ID)*CGRI(ID)
     IF TAC GT TA3
      LET 3CL = (TAC-TA3)*MSR(ID) + (TIME.V-TAC)*MSR(ID)*CGRI(ID)
     ALWAYS
     IF 3CL GT CCL
      LET 3CL = CCL + (3CL-CCL)*(MFR(ID)/MSR(ID))
     ALWAYS
    ALWAYS
   ALWAYS
      LET CL = 1CL+2CL+3CL
      LET S = 2.75 - CL*(1.75/FSAF.LGT)
     IF CL GT FSAF.LGT
      LET S = (LGHT.TO.FAILURE-CL)/(LGHT.TO.FAILURE-FSAF.LGT)
     ALWAYS
      PRINT 2 LINES WITH CL, S/2.75 AS FOLLOWS
```

```
                  SUM OF CRACK LENGTHS AT FAILURE = **.** INCHES
                  RESIDUAL STRENGTH AT FAILURE = .** ULTIMATE
         LEAVE
       ELSE
         LOOP
      ALWAYS
         SCHEDULE AN INCREASE.INSPECTION.FREQUENCY NOW
         SCHEDULE AN IMMEDIATE.FLEET.INSPECTION NOW
         LET FSH(ID) = "NO"
        IF NSFL LT 10
         LET NSFL = NSFL + 1
         LET ACID(NSFL) = ID
         LET FLTHR(NSFL) = TIME.V - ENTRY.TIME(AIRPLANE(ID))
        REGARDLESS
  ''
        IF SNSFL LT 100
         LET SNSFL = SNSFL + 1
         LET SACID(SNSFL) = ID
         LET SFLTHR(SNSFL) = TIME.V - ENTRY.TIME(AIRPLANE(ID))
         LET SSTAN(SNSFL) = ELEMENT(4)
        REGARDLESS
         FOR I = 1 TO 10
         DO
       IF HI.TIME.ACRFT(I) = ID
         ADD 1 TO LHTA
       HERE
        IF LHTA GT SIZE.OF.FLEET
         JUMP AHEAD
        ELSE
         FOR EVERY AIRCRAFT IN CRASHED.FLEET
         DO
        IF LHTA = TAIL.ID
         ADD 1 TO LHTA
         JUMP BACK
        OTHERWISE
         LOOP
        HERE
         LET HOLD = HI.TIME.ACRFT(10)
         LET HI.TIME.ACRFT(10) = LHTA
        IF I GE 10
         LET HI.TIME.ACRFT(I) = HOLD
        REGARDLESS
         LEAVE
      OTHERWISE
         LOOP
     IF SAMP.PENDING(ID) = "NO"
         LET BEEN.MODIFIED = BEEN.MODIFIED + 1
      IF OCCUR.MOD(ID) GT ENTRY.TIME(AIRPLANE(ID))
         LET AIRFRAME.TIME = AIRFRAME.TIME + TIME.V - OCCUR.MOD(ID)
         JUMP AHEAD
      OTHERWISE
         LET AIRFRAME.TIME = AIRFRAME.TIME + TIME.V - ENTRY.TIME(AIRPLANE(ID))
      HERE
```

```
   REGARDLESS
      REMOVE AIRPLANE(ID) FROM ACTIVE.FLEET
      FILE AIRPLANE(ID) IN CRASHED.FLEET
      ADD 1 TO 2.NUM.OF.CRASH
''
'' CANCEL SCHEDULED EVENTS AND DESTROY EVENT NOTICES
''
   IF IE1(ID) = "YES"
      LET 1.ITE = A1E(ID)
      CANCEL THE 1.ITE
      DESTROY THE 1.ITE
    IF IE2(ID) = "YES"
      LET 2.ITE = A2E(ID)
      CANCEL THE 2.ITE
      DESTROY THE 2.ITE
     IF IE3(ID) = "YES"
      LET 3.ITE = A3E(ID)
      CANCEL THE 3.ITE
      DESTROY THE 3.ITE
     ALWAYS
    ALWAYS
   ALWAYS
    IF CO.EXISTS(ID) NE "NO"
      LET COROSION = AC(ID)
     IF CO.EXISTS(ID) = "NS"
      CANCEL THE COROSION
     ALWAYS
      DESTROY THE COROSION
      LET CO.EXISTS(ID) = "NO"
    REGARDLESS
  IF 1.CR.EXISTS(ID) NE "NO"
      LET 1.STRENGTH.REDUCTION = A1SR(ID)
     IF 1.CR.EXISTS(ID) = "NS"
      CANCEL THE 1.STRENGTH.REDUCTION
     ALWAYS
      DESTROY THE 1.STRENGTH.REDUCTION
      LET 1.CR.EXISTS(ID) = "NO"
   IF 2.CR.EXISTS(ID) NE "NO"
      LET 2.STRENGTH.REDUCTION = A2SR(ID)
     IF 2.CR.EXISTS(ID) = "NS"
      CANCEL THE 2.STRENGTH.REDUCTION
     ALWAYS
      DESTROY THE 2.STRENGTH.REDUCTION
      LET 2.CR.EXISTS(ID) = "NO"
    IF 3.CR.EXISTS(ID) NE "NO"
      LET 3.STRENGTH.REDUCTION = A3SR(ID)
     IF 3.CR.EXISTS(ID) = "NS"
      CANCEL THE 3.STRENGTH.REDUCTION
     ALWAYS
      DESTROY THE 3.STRENGTH.REDUCTION
      LET 3.CR.EXISTS(ID) = "NO"
    REGARDLESS
   REGARDLESS
```

```
    REGARDLESS
      IF AIL(ID) = "YES"
        LET REACH.FAIL.SAFE.LGT = ARFSL(ID)
        CANCEL THE REACH.FAIL.SAFE.LGT
        DESTROY THE REACH.FAIL.SAFE.LGT
        LET AIL(ID) = "NO"
      REGARDLESS
        LET D.LEVEL.INSPECTION = ADL(ID)
        CANCEL THE D.LEVEL.INSPECTION
        DESTROY THE D.LEVEL.INSPECTION
      IF INSP.SCH(ID) = "YES"
        CALL CANCEL.SCHEDULED.INSPECTIONS
      REGARDLESS
 ''
        FOR EACH RETIRE.FROM.SERVICE IN EV.S(I.RETIRE.FROM.SERVICE)
        DO
        IF IORET = ID        CANCEL THE RETIRE.FROM.SERVICE
        DESTROY THE RETIRE.FROM.SERVICE
        LEAVE          ELSE
        LOOP
        FOR EACH IN.SERVICE.DAMAGE IN EV.S(I.IN.SERVICE.DAMAGE)
        DO
        IF IDSD  = ID        CANCEL THE IN.SERVICE.DAMAGE
        DESTROY THE IN.SERVICE.DAMAGE
        LEAVE          ELSE
        LOOP
        FOR EACH T.INSPECTION.INCREASE IN EV.S(I.T.INSPECTION.INCREASE)
        DO
        IF IDTI = ID         CANCEL THE T.INSPECTION.INCREASE
        DESTROY THE T.INSPECTION.INCREASE
        LEAVE    ELSE
        LOOP
        RETURN
        END


           LOCAL VARIABLES OF THIS ROUTINE

CCL        DOUBLE   WORD 19      CL         DOUBLE   WORD 33
HOLD       INTEGER  WORD  3      I.1        INTEGER  WORD  7
I.2        INTEGER  WORD  8      I.3        INTEGER  WORD  9
IDEA       INTEGER  WORD  1      J.1        INTEGER  WORD 10
K.1        INTEGER  WORD 12      K.2        INTEGER  WORD 13
K.3        INTEGER  WORD 14      K.4        INTEGER  WORD 15
L.12       INTEGER  WORD 37      L.16       INTEGER  WORD 38
L.20       INTEGER  WORD 39      L.24       INTEGER  WORD 40
L.4        INTEGER  WORD  4      M.1        INTEGER  WORD 11
R.1        DOUBLE   WORD  5      S          DOUBLE   WORD 35
TAC        DOUBLE   WORD 21      TA1        DOUBLE   WORD 17
TA2        DOUBLE   WORD 25      TA3        DOUBLE   WORD 29
1CL        DOUBLE   WORD 23      2CL        DOUBLE   WORD 27
3CL        DOUBLE   WORD 31
```

```
EVENT RETIRE.FROM.SERVICE(IDRET)
''
'' REPRESENTS RETIREMENT OF THE AIRCRAFT FROM SERVICE
''
      DEFINE IDRET AS AN INTEGER VARIABLE
      DEFINE HOLD AS AN INTEGER VARIABLE
      LET ID = IDRET
   IF LIST = "YES"
      FOR I = 1 TO NOACC(LOX)
      DO
    IF ID = TLIO(LOX,I)
      SKIP 1 OUTPUT LINE
      PRINT 1 LINE WITH ID, TIME.V-ENTRY.TIME(AIRPLANE(ID)) AS FOLLOWS
        A/C **, *** RETIRED FROM SERVICE AT ****** FLIGHT HOURS
      LEAVE
   ELSE
      LOOP
  ALWAYS
     FOR J = 1 TO 10
     DO
   IF HI.TIME.ACRFT(I) = ID
     ADD 1 TO LHTA
   HERE
    IF LHTA GT SIZE.OF.FLEET
     JUMP AHEAD
    ELSE
    FOR EVERY AIRCRSET I. CRASHED.FLEET
     DO
    IF LHTA = TAIL.ID
     ADD 1 TO LHTA
     JUMP BACK
   OTHERWISE
    LOOP
    LET HOLD = HI.TIME.ACRFT(10)
    LET HI.TIME.ACRFT(10) = LHTA
   IF I NE J
    LET HI.TIME.ACRFT(I) = HOLD
   REGARDLESS
    LEAVE
  OTHERWISE
    LOOP
   HERE
  IF [illegible] = [illegible]
    LET NEW.MODIFIER = ITEM.MODIFIER + 1
  IF OCCUR.ADD(ID) GT E.ENTRY.TIME(AIRPLANE(ID))
   LET AIRFRAME.TIME = AIRFRAME.TIME + TIME.V - OCCUR.MOD(ID)
   JUMP AHEAD
  OTHERWISE
   LET AIRFRAME.TIME = AIRFRAME.TIME + TIME.V - ENTRY.TIME(AIRPLANE(ID))
 HERE
 REGARDLESS
   REMOVE AIRPLANE(ID) FROM ACTIVE.FLEET
   FILE AIRPLANE(ID) IN FLEET.RETIRED
```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
     ADD 1 TO N.TO.BE.RETIRE
''
'' CANCEL SCHEDULED EVENTS AND DESTROY EVENT NOTICES
''
  IF IE1(ID) = "YES"
     LET 1.ITE = A1E(ID)
     CANCEL THE 1.ITE
     DESTROY THE 1.ITE
   IF IE2(ID) = "YES"
     LET 2.ITE = A2E(ID)
     CANCEL THE 2.ITE
     DESTROY THE 2.ITE
    IF IE3(ID) = "YES"
     LET 3.ITE = A3E(ID)
     CANCEL THE 3.ITE
     DESTROY THE 3.ITE
    ALWAYS
   ALWAYS
  ALWAYS
   IF CO.EXISTS(ID) NE "NO"
     LET CORROSION = AC(ID)
    IF CO.EXISTS(ID) = "YS"
     CANCEL THE CORROSION
    ALWAYS
     DESTROY THE CORROSION
     LET CO.EXISTS(ID) = "NO"
   REGARDLESS
 IF 1.CH.EXISTS(ID) NE "NN"
     LET 1.STRENGTH.REDUCTION = A1SR(ID)
    IF 1.CH.EXISTS(ID) = "YS"
     CANCEL THE 1.STRENGTH.REDUCTION
    ALWAYS
     DESTROY THE 1.STRENGTH.REDUCTION
     LET 1.CH.EXISTS(ID) = "NN"
  IF 2.CH.EXISTS(ID) NE "NN"
     LET 2.STRENGTH.REDUCTION = A2SR(ID)
    IF 2.CH.EXISTS(ID) = "YS"
     CANCEL THE 2.STRENGTH.REDUCTION
    ALWAYS
     DESTROY THE 2.STRENGTH.REDUCTION
     LET 2.CH.EXISTS(ID) = "NN"
   IF 3.CH.EXISTS(ID) NE "NN"
     LET 3.STRENGTH.REDUCTION = A3SR(ID)
    IF 3.CH.EXISTS(ID) = "YS"
     CANCEL THE 3.STRENGTH.REDUCTION
    ALWAYS
     DESTROY THE 3.STRENGTH.REDUCTION
     LET 3.CH.EXISTS(ID) = "NN"
   REGARDLESS
  REGARDLESS
 REGARDLESS
   IF INSP.SCH(ID) = "YES"
     CALL CANCEL.SCHEDULED.INSPECTIONS
```

```
LINE  CACI SIMSCRIPT II.5  RELEASE 8F                              09/01/76  PAGE  68

          REGARDLESS
            LET D.LEVEL.INSPECTION = ADL(ID)
            CANCEL THE D.LEVEL.INSPECTION
            DESTROY THE D.LEVEL.INSPECTION
          IF AIL(ID) = "YES"
            LET REACH.FAIL.SAFE.LGT = ARFSL(ID)
            CANCEL THE REACH.FAIL.SAFE.LGT
            DESTROY THE REACH.FAIL.SAFE.LGT
            LET AIL(ID) = "NO"
          REGARDLESS
          IF FSH(ID) = "YES"
            LET FAILURE = AF(ID)
            CANCEL THE FAILURE
            DESTROY THE FAILURE
            LET FSH(ID) = "NO"
          ALWAYS
     ''
            RETURN
            END

                    LOCAL VARIABLES OF THIS ROUTINE

HOLD        INTEGER  WORD  3        I.1         INTEGER  WORD  7
I.2         INTEGER  WORD  8        I.3         INTEGER  WORD  9
IORET       INTEGER  WORD  1        J.1         INTEGER  WORD 10
K.1         INTEGER  WORD 12        K.2         INTEGER  WORD 13
K.3         INTEGER  WORD 14        K.4         INTEGER  WORD 15
L.12        INTEGER  WORD 16        L.4         INTEGER  WORD  4
N.1         INTEGER  WORD 11        R.1         DOUBLE   WORD  5
```

```
EVENT REPAIR(IOREP)
      DEFINE IOREP AS AN INTEGER VARIABLE
      LET ID = IOREP
   IF TMOD.PENDING(ID) = "YES" OR SMOD.PENDING(ID) = "YES"
    IF TMOD.PENDING(ID) = "YES"
      CALL INSTALL.MODIFICATION
      LET TMOD.PENDING(ID) = "NO"
     IF TIME.V LT ENTRY.TIME(AIRPLANE(ID)) + C7 * 144EL
      LET T.INSPECTION.INCREASE = ATII(ID)
      CANCEL THE T.INSPECTION.INCREASE
      DESTROY THE T.INSPECTION.INCREASE
     REGARDLESS
      RETURN
    OTHERWISE
      CALL INSTALL.MODIFICATION
      LET SMOD.PENDING(ID) = "NO"
      RETURN
   OTHERWISE
 ''
     IF FSH(ID) = "YES"
      LET FAILURE = AF(ID)
      CANCEL THE FAILURE
      DESTROY THE FAILURE
      LET FSH(ID) = "NO"
     ALWAYS
     IF AIL(ID) = "YES"
      LET REACH.FAIL.SAFE.LGT = ARFSL(ID)
      CANCEL THE REACH.FAIL.SAFE.LGT
      DESTROY THE REACH.FAIL.SAFE.LGT
      LET AIL(ID) = "NO"
     REGARDLESS
      LET RST = USAGE.LIFE - TIME.V + ENTRY.TIME(AIRPLANE(ID))
     IF CO.EXISTS(ID) = "YES"
      LET CGRI(ID) = 1.0
      LET CORROSION = AC(ID)
      LET TAC = TIME.A(CORROSION)
      DESTROY THE COROSION
      LET CO.EXISTS(ID) = "NO"
      CALL PREDICT.CORROSION YIELDING HOURS.TO.CORROSION
      IF HOURS.TO.CORROSION LT RST
      SCHEDULE A CORROSION(ID) AT TIME.V + HOURS.TO.CORROSION
      LET AC(ID) = CORROSION
      LET CORRP.TIME(ID) = TIME.V
      LET CO.EXISTS(ID) = "NS"
     REGARDLESS
    REGARDLESS
   IF 1.CR.EXISTS(ID) = "YES"
      LET CCL = CRIT.CRK.LGT
    IF CHG.FREQ.TIME = TIME.V
      JUMP AHEAD
    OTHERWISE
      LET 1.STRENGTH.REDUCTION = AISR(ID)
      LET TA1 = TIME.A(1.STRENGTH.REDUCTION)
```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
     LET CL = (TIME.V-TA1) * MSR(ID) * CGRI(ID)
   IF TAC GT TA1
     LET CL = (TAC-TA1)*MSR(ID) + (TIME.V-TAC)*MSR(ID)*CGRI(ID)
   REGARDLESS
   IF CCL LT CL
     LET CL = CCL + ((CL-CCL)/MSR(ID))*MFR(ID)
   REGARDLESS
     LET MAX.CRK = CL
     LET STR.RED = CL
  IF 2.CR.EXISTS(ID) = "YES"
     LET 2.STRENGTH.REDUCTION = A2SR(ID)
     LET TA2 = TIME.A(2.STRENGTH.REDUCTION)
     LET CL = (TIME.V-TA2) * MSR(ID) * CGRI(ID)
   IF TAC GT TA2
     LET CL = (TAC-TA2)*MSR(ID) + (TIME.V-TAC)*MSR(ID)*CGRI(ID)
   REGARDLESS
   IF CCL LT CL
     LET CL = CCL + ((CL-CCL)/MSR(ID))*MFR(ID)
   REGARDLESS
   IF MAX.CRK LT CL
     LET MAX.CRK = CL
   REGARDLESS
     ADD CL TO STR.RED
   IF 3.CR.EXISTS(ID) = "YES"
     LET 3.STRENGTH.REDUCTION = A3SR(ID)
     LET TA3 = TIME.A(3.STRENGTH.REDUCTION)
     LET CL = (TIME.V-TA3) * MSR(ID) * CGRI(ID)
    IF TAC GT TA3
      LET CL = (TAC-TA2)*MSR(ID) + (TIME.V-TAC)*MSR(ID)*CGRI(ID)
    REGARDLESS
    IF CCL LT CL
      LET CL = CCL + ((CL-CCL)/MSR(ID))*MFR(ID)
    REGARDLESS
    IF MAX.CRK LT CL
      LET MAX.CRK = CL
    REGARDLESS
      ADD CL TO STR.RED
   REGARDLESS
  REGARDLESS
     LET FLEET.STR.RED = FLEET.STR.RED + STR.RED
     LET POT.CRK = MAX.CRK + M1.MEAN * AHCR(LIL)
   IF POT.CRK GT CCL
     LET POT.CRK = CCL + ((POT.CRK-CCL)/M1.MEAN) * M2.MEAN
   REGARDLESS
   IF MAX.CRK GT CCL
     LET POT.CRK = MAX.CRK + M2.MEAN * AHCR(LIL)
   REGARDLESS
   IF POT.CRK + STR.RED - MAX.CRK GT FSAF.LGT/2.0
     SCHEDULE AN INCREASE.INSPECTION.FREQUENCY NOW
     SCHEDULE AN IMMEDIATE.FLEET.INSPECTION NOW
     LET CHG.FREQ.TIME = TIME.V
     JUMP AHEAD
   REGARDLESS
```

```
      IF FLEET.STR.RED GT (FSAF.LGT/5.0) * (IDCK-1.NUM.OF.RETIRE-2.NUM.OF.CRASH)
        SCHEDULE AN INCREASE.INSPECTION.FREQUENCY NOW
        SCHEDULE AN IMMEDIATE.FLEET.INSPECTION NOW
        LET CHG.FREQ.TIME = TIME.V
      REGARDLESS
      HERE
  ''
        LET COST.OF.REPAIRS = COST.OF.REPAIRS + FIXIT.COST
     IF TMP.MOD.SCH = "NO" AND DEC.ON.MOD.SCH = "NO"
      IF TES.FAILURE = "YES" AND TIME.V LT START.TEST + ACTUAL.AVG.FAT.LIFE /
        TEST.ACCEL.FACT OR TES.FAILURE = "YES" AND TIME.V GT START.TEST +
        ACTUAL.AVG.FAT.LIFE / TEST.ACCEL.FACT + LEAD.TIME OR TES.FAILURE = "NO"
       IF TO.BE.MODIFIED - BEEN.MODIFIED - 1.NUM.OF.RETIRE - 2.NUM.OF.CRASH LE 0
        SCHEDULE A DECISION.ON.MOD NOW
        LET DEC.ON.MOD.SCH = "YES"
       REGARDLESS
      REGARDLESS
     REGARDLESS
  ''
        LET AAFL = ACTUAL.AVG.FAT.LIFE
      IF I LE FDCK
        LET AAFL = IAAFL
      REGARDLESS
        CALL FATIGUE.LIFE.SCATTER(AAFL,3) YIELDING FIRST.LIFE, SECOND.LIFE AND
        THIRD.LIFE
        LET 1.STRENGTH.REDUCTION = A1SR(ID)
        LET 2.STRENGTH.REDUCTION = A2SR(ID)
        LET 3.STRENGTH.REDUCTION = A3SR(ID)
        DESTROY THE 1.STRENGTH.REDUCTION
     IF 2.CR.EXISTS(ID) = "YES"
        DESTROY THE 2.STRENGTH.REDUCTION
      IF 3.CR.EXISTS(ID) = "YES"
        DESTROY THE 3.STRENGTH.REDUCTION
        JUMP AHEAD
      OTHERWISE
      IF 3.CR.EXISTS(ID) = "NS"
       IF TIME.A(3.STRENGTH.REDUCTION) - TIME.V LT FIRST.LIFE
        LET THIRD.LIFE = SECOND.LIFE
        LET SECOND.LIFE = FIRST.LIFE
        LET FIRST.LIFE = TIME.A(3.STRENGTH.REDUCTION) - TIME.V
        CANCEL THE 3.STRENGTH.REDUCTION
        DESTROY THE 3.STRENGTH.REDUCTION
        JUMP AHEAD
       ELSE
       IF TIME.A(3.STRENGTH.REDUCTION) - TIME.V LT SECOND.LIFE
        LET THIRD.LIFE = SECOND.LIFE
        LET SECOND.LIFE = TIME.A(3.STRENGTH.REDUCTION) - TIME.V
        CANCEL THE 3.STRENGTH.REDUCTION
        DESTROY THE 3.STRENGTH.REDUCTION
        JUMP AHEAD
       ELSE
        LET THIRD.LIFE = TIME.A(3.STRENGTH.REDUCTION) - TIME.V
        CANCEL THE 3.STRENGTH.REDUCTION
```

```
   DESTROY THE 3.STRENGTH.REDUCTION
   JUMP AHEAD
 OTHERWISE
   LET THIRD.LIFE = USAGE.LIFE
   JUMP AHEAD
OTHERWISE
IF 2.CR.EXISTS(ID) = "NS"
 IF 3.CR.EXISTS(ID) = "NS"
  IF TIME.A(3.STRENGTH.REDUCTION).- TIME.V LT FIRST.LIFE
   LET THIRD.LIFE = FIRST.LIFE
   LET SECOND.LIFE = TIME.A(3.STRENGTH.REDUCTION) - TIME.V
   LET FIRST.LIFE = TIME.A(2.STRENGTH.REDUCTION) - TIME.V
   CANCEL THE 2.STRENGTH.REDUCTION
   DESTROY THE 2.STRENGTH.REDUCTION
   CANCEL THE 3.STRENGTH.REDUCTION
   DESTROY THE 3.STRENGTH.REDUCTION
   JUMP AHEAD
  OTHERWISE
  IF TIME.A(2.STRENGTH.REDUCTION) - TIME.V LT FIRST.LIFE
   LET THIRD.LIFE = TIME.A(3.STRENGTH.REDUCTION) - TIME.V
   LET SECOND.LIFE = FIRST.LIFE
   LET FIRST.LIFE = TIME.A(2.STRENGTH.REDUCTION) - TIME.V
   CANCEL THE 2.STRENGTH.REDUCTION
   DESTROY THE 2.STRENGTH.REDUCTION
   CANCEL THE 3.STRENGTH.REDUCTION
   DESTROY THE 3.STRENGTH.REDUCTION
   JUMP AHEAD
  OTHERWISE
   LET SECOND.LIFE = TIME.A(2.STRENGTH.REDUCTION)
   LET THIRD.LIFE = TIME.A(3.STRENGTH.REDUCTION)
   CANCEL THE 2.STRENGTH.REDUCTION
   DESTROY THE 2.STRENGTH.REDUCTION
   CANCEL THE 3.STRENGTH.REDUCTION
   DESTROY THE 3.STRENGTH.REDUCTION
   JUMP AHEAD
 OTHERWISE
 IF TIME.A(2.STRENGTH.REDUCTION) - TIME.V LT FIRST.LIFE
   LET THIRD.LIFE = USAGE.LIFE
   LET SECOND.LIFE = FIRST.LIFE
   LET FIRST.LIFE = TIME.A(2.STRENGTH.REDUCTION)
   CANCEL THE 2.STRENGTH.REDUCTION
   DESTROY THE 2.STRENGTH.REDUCTION
   JUMP AHEAD
 OTHERWISE
   LET SECOND.LIFE = TIME.A(2.STRENGTH.REDUCTION)
   JUMP AHEAD
OTHERWISE
   LET SECOND.LIFE = USAGE.LIFE
   LET THIRD.LIFE = USAGE.LIFE
 HERE
   LET 1.CR.EXISTS(ID) = "NN"
   LET 2.CR.EXISTS(ID) = "NN"
   LET 3.CR.EXISTS(ID) = "NN"
```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
    IF FIRST.LIFE LT RST OR CO.EXISTS(ID) = "NS" OR SD.SCH(ID) = "YES"
       SCHEDULE A 1.STRENGTH.REDUCTION(ID) AT TIME.V + FIRST.LIFE
       LET A1SR(ID) = 1.STRENGTH.REDUCTION
       LET 1.CR.EXISTS(ID) = "NS"
     IF SECOND.LIFE LT RST OR CO.EXISTS(ID) = "NS" OR SD.SCH(ID) = "YES"
       SCHEDULE A 2.STRENGTH.REDUCTION(ID) AT TIME.V + SECOND.LIFE
       LET A2SR(ID) = 2.STRENGTH.REDUCTION
       LET 2.CR.EXISTS(ID) = "NS"
      IF THIRD.LIFE LT RST OR CO.EXISTS(ID) = "NS" OR SD.SCH(ID) = "YES"
       SCHEDULE A 3.STRENGTH.REDUCTION(ID) AT TIME.V + THIRD.LIFE
       LET A3SR(ID) = 3.STRENGTH.REDUCTION
       LET 3.CR.EXISTS(ID) = "NS"
      REGARDLESS
     REGARDLESS
    REGARDLESS
   REGARDLESS
    IF IF1(ID) = "YES"
       LET 1.ITE = A1E(ID)
       LET IF1(ID) = "NO"
       CANCEL THE 1.ITE
       DESTROY THE 1.ITE
     IF IF2(ID) = "YES"
       LET 2.ITE = A2E(ID)
       LET IF2(ID) = "NO"
       CANCEL THE 2.ITE
       DESTROY THE 2.ITE
      IF IE3(ID) = "YES"
       LET 3.ITE = A3E(ID)
       LET IF3(ID) = "NO"
       CANCEL THE 3.ITE
       DESTROY THE 3.ITE
      ALWAYS
     ALWAYS
    ALWAYS
    IF LTH1 = "YES"
       FOR I = 1 TO NOAC(LDX)
       DO
     IF ID = TLID(LDX,I)
       SKIP 1 OUTPUT LINE
       PRINT 4 LINES WITH ID, TIME.V-ENTRY.TIME(AIRPLANE(ID)), FIRST.LIFE,
       SECOND.LIFE, THIRD.LIFE AS FOLLOWS
         A/C NO. *** HAS ALL DEFECTS REPAIRED AT ****** FLIGHT HOURS
              1ST CRACK INITIATION PROJECTED AT ******* FLIGHT HOURS
              2ND CRACK INITIATION PROJECTED AT ******* FLIGHT HOURS
              3RD CRACK INITIATION PROJECTED AT ******* FLIGHT HOURS
       LEAVE
     ELSE
       LOOP
    ALWAYS
 ''
 '' CANCEL SCHEDULED INSPECTIONS ON THIS AIRCRAFT
 ''
          CALL CANCEL.SCHEDULED.INSPECTIONS
```

```
LINE  CACI SIMSCRIPT II.5  RELEASE 8F                              09/01/76  PAGE  74

 262         RETURN
 263         END

                   LOCAL VARIABLES OF THIS ROUTINE

 AAFL        DOUBLE     WORD 25       CCL         DOUBLE     WORD  9
 CL          DOUBLE     WORD 13       FIRST.LIFE  DOUBLE     WORD 27
 HOURS.TO.C  DOUBLE     WORD  7       I.1         INTEGER    WORD 37
 I.2         INTEGER    WORD 38       I.3         INTEGER    WORD 39
 IOREP       INTEGER    WORD  1       J.1         INTEGER    WORD 40
 K.1         INTEGER    WORD 42       K.2         INTEGER    WORD 43
 K.3         INTEGER    WORD 44       K.4         INTEGER    WORD 45
 L.4         INTEGER    WORD 33       MAX.CRK     DOUBLE     WORD 15
 N.1         INTEGER    WORD 41       OPT.CRK     DOUBLE     WORD 23
 R.1         DOUBLE     WORD 35       RST         DOUBLE     WORD  3
 SECOND.LIF  DOUBLE     WORD 29       STR.RED     DOUBLE     WORD 17
 TAC         DOUBLE     WORD  5       TA1         DOUBLE     WORD 11
 TA2         DOUBLE     WORD 19       TA3         DOUBLE     WORD 21
 THIRD.LIFE  DOUBLE     WORD 31
```

```
EVENT T.INSPECTION.INCREASE(IDTI)
         DEFINE IDTI AS AN INTEGER VARIABLE
         LET ID = IDTI
       IF LID = 3 OR LEL = 3
         LET C.INTERVAL(ID) = C.INTERVAL(ID) * T.FREQ.CHG
       REGARDLESS
''
       IF ID = 4 OR LEL = 4
         LET D.INTERVAL(ID) = D.INTERVAL(ID) * T.FREQ.CHG
       REGARDLESS
         RETURN
         END
```

```
              LOCAL VARIABLES OF THIS ROUTINE

IDTI         INTEGER   WORD  1
```

```
EVENT INCREASE.INSPECTION.FREQUENCY
''
''  INCREASES THE FREQUENCY OF THE LOWEST LEVEL INTERNAL AND EXTERNAL INSPECTIONS
''     THESE MAY OR MAY NOT BE THE SAME LEVEL OF INSPECTION
''     THE A-LEVEL AND B-LEVEL FREQUENCIES ARE NEVER CHANGED
''
      LET FLEET.STR.RED = 0.0
 IF ARCD(LIL)*S.FREQ.CHG GE 1ARCD(LIL-1) OR ARCD(LEL)*S.FREQ.CHG GE 1ARCD(LEL-1)
     IF ARCD(LIL)*S.FREQ.CHG GE 1ARCD(LIL-1)
      LET DIFF = ARCD(LIL) - ARCD(LIL) * S.FREQ.CHG
      LET ARCD(LIL) = ARCD(LIL) * S.FREQ.CHG
     ALWAYS
    IF LIL NE LEL
     IF ARCD(LEL)*S.FREQ.CHG GE 1ARCD(LEL-1)
      LET ARCD(LEL) = ARCD(LEL) * S.FREQ.CHG
     ALWAYS
    REGARDLESS
     IF LTHO = "YES"
      SKIP 1 OUTPUT LINE
      PRINT 3 LINES WITH ARCD(3), ARCD(4) AS FOLLOWS
        INSPECTION INTERVAL DECREASE IMPLEMENTED
            C-LEVEL INTERVAL NOW ***** HOURS
            D-LEVEL INTERVAL NOW ***** HOURS
     ALWAYS
      LET CINSL = ARCD(3)
      LET DINSL = ARCD(4)
      LET NICHG = NICHG +1
     IF NICHG GT 30
      JUMP AHEAD
    OTHERWISE
      LET SC(NICHG) = ARCD(3)
      LET SD(NICHG) = ARCD(4)
    HERE
      FOR EVERY AIRCRAFT IN ACTIVE.FLEET
      DO
      LET ID = TAIL.ID
      LET C.INTERVAL(ID) = ARCD(3)
      LET D.INTERVAL(ID) = ARCD(4)
    IF LIL = 4
      LET D.LEVEL.INSPECTION = ADL(ID)
      CANCEL THE D.LEVEL.INSPECTION
     IF TIME.A(D.LEVEL.INSPECTION) - TIME.V LT DIFF
      RESCHEDULE THE D.LEVEL.INSPECTION(ID) NOW
       JUMP AHEAD
     OTHERWISE
      RESCHEDULE THE D.LEVEL.INSPECTION(ID) AT TIME.A(D.LEVEL.INSPECTION) - DIFF
       JUMP AHEAD
    OTHERWISE
 ''
    IF INSP.SCH(ID) = "YES"
       LET C.LEVEL.INSPECTION = ACL(ID)
       CANCEL THE C.LEVEL.INSPECTION
     IF TIME.A(C.LEVEL.INSPECTION) - TIME.V LT DIFF
```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
                                                                  09/01/76  PAGE  77
LINE  CACI SIMSCRIPT II.5  RELEASE 8F
 51          RESCHEDULE THE C.LEVEL.INSPECTION(ID) NOW
 52            JUMP AHEAD
 53          OTHERWISE
 54            RESCHEDULE THE C.LEVEL.INSPECTION(ID) AT TIME.A(C.LEVEL.INSPECTION) - DIFF
 55         REGARDLESS
 56          HERE
 57            LOOP
 58       ALWAYS
 59            RETURN
 60            END

                  LOCAL VARIABLES OF THIS ROUTINE

DIFF            DOUBLE     WORD  1        I.1          INTEGER   WORD  7
I.2             INTEGER    WORD  8        I.3          INTEGER   WORD  9
J.1             INTEGER    WORD 10        K.1          INTEGER   WORD 12
K.2             INTEGER    WORD 13        K.3          INTEGER   WORD 14
K.4             INTEGER    WORD 15        L.1          INTEGER   WORD  3
L.2             INTEGER    WORD 16        M.1          INTEGER   WORD 11
M.1             DOUBLE     WORD  5
```

```
EVENT IMMEDIATE.FLEET.INSPECTION
''
'' THIS EVENT REPRESENTS AN IMMEDIATE FLEET WIDE INSPECTION CAUSED BY FINDING A
''    DEFECT CONSIDERED TOO HAZARDOUS TO DEPEND ON SCHEDULED INSPECTIONS
''
'' THIS EVENT IS ALWAYS PRECEDED BY THE EVENT INCREASE.INSPECTION.FREQUENCY
''
      IF LTHO = "YES"
       SKIP 1 OUTPUT LINE
       PRINT 1 LINE AS FOLLOWS
         FLEET WIDE SPECIAL INSPECTION PERFORMED
      ALWAYS
       LET CCL = CRIT.CHK.LGT
       LET NSIC = NSIC + 1
       LET FIXIT.COST = S.REPAIR.COST
       FOR EVERY AIRCRAFT IN ACTIVE.FLEET
       DO
       LET LIST = 0.0
       LET ID = TAIL.ID
     IF LTHO = "YES"
       FOR I = 1 TO NOAC(LDX)
       DO
      IF ID = ILTO(LDX,I)
       LET LIST = 1.0
       LEAVE
     ELSE
       LOOP
     ALWAYS
       LET M1 = MSM(ID)
       LET M2 = MRR(ID)
       LET FOUND = 0
       LET TAC = 0.0
''
     IF CO.EXISTS(ID) = "YES"
       LET CORROSION = FC(ID)
       LET TAC = TIME.A(CORROSION)
       LET AREA = C.GROWTH.RATE * CRRF * (TIME.V - TAC)
      IF RANDOM.F(1) LE PODD(.999,.971,.180,AREA)
       IF LIST = 1.0
       SKIP 1 OUTPUT LINE
       PRINT 1 DOUBLE LINE WITH AREA, ID, TIME.V+ENTRY.TIME(AIRPLANE(ID)) AS
       FOLLOWS
         CORROSION OF AREA **.** SQ. INCHES DETECTED DURING SPECIAL INSPECTION OF A/C NO. *** AT ****** FLIGHT HOURS
       ALWAYS
       LET SCA = AREA
       LET TSCA = AREA
       LET NSCD = NSCD + 1
       LET SNSCD = SNSCD + 1
       LET FOUND = 2
      REGARDLESS
     REGARDLESS
''
    IF 1.TH.EXISTS(ID) = "YES"
```

```
         LET 1.STRENGTH.REDUCTION = A1SH(ID)
         LET TA1 = TIME.AT(1.STRENGTH.REDUCTION)
         LET CL = (TIME.V-TA1) * M1 * CGRI(ID)
       IF TAC GT TA1
         LET CL = (TAC-TA1)*M1 + (TIME.V-TAC)*M1*CGRI(ID)
       REGARDLESS
       IF CL GT CCL
         LET CL = CCL + ((CL-CCL)/M1)*M2
       REGARDLESS
         LET SM = CL
         LET MAX.CRK = CL
   ''
       IF RANDOM.F(1) LE PROB(.996,.971,.102,CL)
        IF LIST = 1.0
         SKIP 1 OUTPUT LINE
         PRINT 1 DOUBLE LINE WITH CL, ID, TIME.V, REENTRY.TIME(AIRPLANE(ID)) AS
         FOLLOWS
           CRACK OF LENGTH **.** INCHES DETECTED DURING SPECIAL INSPECTION OF A/C NO. *** AT ****** FLIGHT HOURS
        ALWAYS
         LET SCRK = CL
         LET GSCRK = CL
         ADD 1 TO NSCR
         ADD 1 TO GNSCR
         LET FOUND = 1
       REGARDLESS
   ''
      IF 2.CR.EXISTS(ID) = "YES"
         LET 2.STRENGTH.REDUCTION = A2SH(ID)
         LET TA2 = TIME.AT(2.STRENGTH.REDUCTION)
         LET CL = (TIME.V-TA2) * M1 * CGRI(ID)
       IF TAC GT TA2
         LET CL = (TAC-TA2)*M1 + (TIME.V-TAC)*M1*CGRI(ID)
       REGARDLESS
       IF CL GT CCL
         LET CL = CCL + ((CL-CCL)/M1)*M2
       REGARDLESS
         ADD CL TO SM
       IF MAX.CRK LT CL
         LET MAX.CRK = CL
       REGARDLESS
   ''
       IF RANDOM.F(1) LE PROB(.996,.971,.102,CL)
        IF LIST = 1.0
         SKIP 1 OUTPUT LINE
         PRINT 1 DOUBLE LINE WITH CL, ID, TIME.V, REENTRY.TIME(AIRPLANE(ID)) AS
         FOLLOWS
           CRACK OF LENGTH **.** INCHES DETECTED DURING SPECIAL INSPECTION OF A/C NO. *** AT ****** FLIGHT HOURS
        ALWAYS
         LET SCRK = CL
         LET GSCRK = CL
         ADD 1 TO NSCR
         ADD 1 TO GNSCR
         LET FOUND = 1
        REGARDLESS
  ''
        IF S.CRK.EXISTS(ID) = "YES"
          LET S.STRENGTH.REDUCTION = B3S(ID)
          LET THIS.LIFE = USAGE.LIFE
          CANCEL THE 2.STRENGTH.REDUCTION
          DESTROY THE 2.STRENGTH.REDUCTION
          LET TAS = TIME.A(S.STRENGTH.REDUCTION)
          LET CL = (TIME.V-TAS) * M1 * CGHI(ID)
         IF TAC GT TAS
          LET CL = (TAC-TAS)*M1 + (TIME.V-TAC)*M1*CGHI(ID)
         REGARDLESS
         IF CL GT CCL
          LET CL = CCL + ((CL-CCL)/M1)**2
         REGARDLESS
          ADD CL TO SR
         IF MAX.CRK LT CL
          LET MAX.CRK = CL
         REGARDLESS
  ''
         IF RS.SIZE.F(1) LE RAND(.999,.971,.102,CL)
         IF LIST = 1.0
          SKIP 1 OUTPUT LINE
          PRINT 1 DOUBLE LINE WITH CL, ID, TIME.V-ENTRY.TIME(AIRPLANE(ID)) AS
          FOLLOWS
             CRACK OF LENGTH **.** INCHES DETECTED DURING SPECIAL INSPECTION OF A/C NO. *** AT ****** FLIGHT HOURS
         ALWAYS
          LET SCRAL = CL
          LET STRK = CL
          ADD 1 TO NSCR
          ADD 1 TO RISCR
          LET FOUND = 1
         REGARDLESS
  ''
       REGARDLESS
      REGARDLESS
     REGARDLESS
  ''
          LET STR.RED = SR
        IF FOUND = 1
          FOR I = 1 TO 10
          DO
         IF RT.TIME.ACRFT(I) = ID
          LET RROD(I) = TIME.V
          LEAVE
         ELSE
          LOOP
        REGARDLESS
       IF FOUND GT 0
          SCHEDULE A REPAIR(ID) NOW
        IF TMOD.PENDING(ID) = "NO" AND SMOD.PENDING(ID) = "NO"
          LET POT.CRK = MAX.CRK + M1.MEAN * AHCD(LIL)
         IF POT.CRK GT CCL
```

```
LINE  CACI SIMSCRIPT II.5  RELEASE 8F                              09/01/76  PAGE  81

            LET POT.CRK = CCL + ((POT.CRK-CCL)/M1.MEAN) * M2.MEAN
           REGARDLESS
           IF MAX.CRK GT CCL
            LET POT.CRK = MAX.CRK + M2.MEAN * ABCD(LIL)
           REGARDLESS
           IF POT.STR.RED LT POT.CRK + STR.RED - MAX.CRK
            LET POT.STR.RED = POT.CRK + STR.RED - MAX.CRK
           REGARDLESS
          REGARDLESS
         REGARDLESS
            LOOP
    ''
    '' DEFECTS FOUND DURING THIS INSPECTION CAN CAUSE AN ADDITIONAL INCREASE IN THE
    ''     FREQUENCY OF NORMALLY SCHEDULED INSPECTIONS
    ''
          IF POT.STR.RED GT ESAF.LGT/2.0
            SCHEDULE AN INCREASE.INSPECTION.FREQUENCY NOW
          REGARDLESS
    ''
            RETURN
            END


                      LOCAL VARIABLES OF THIS ROUTINE

ARFA          DOUBLE    WORD 29        CCL          DOUBLE    WORD 15
CL            DOUBLE    WORD 35        FOUND        DOUBLE    WORD 25
I.1           INTEGER   WORD  5        I.2          INTEGER   WORD  6
I.3           INTEGER   WORD  7        J.1          INTEGER   WORD  8
K.1           INTEGER   WORD 10        K.2          INTEGER   WORD 11
K.3           INTEGER   WORD 12        K.4          INTEGER   WORD 13
L.1           INTEGER   WORD  1        L.10         INTEGER   WORD 31
L.12          INTEGER   WORD 41        L.14         INTEGER   WORD 45
L.16          INTEGER   WORD 51        L.6          INTEGER   WORD 17
LIST          DOUBLE    WORD 19        MAX.CRK      DOUBLE    WORD 39
M1            DOUBLE    WORD 21        M2           DOUBLE    WORD 23
N.1           INTEGER   WORD  9        POT.CRK      DOUBLE    WORD 55
POT.STR.RE    DOUBLE    WORD 57        P.1          DOUBLE    WORD  3
SR            DOUBLE    WORD 37        STR.RED      DOUBLE    WORD 53
TAC           DOUBLE    WORD 27        TA1          DOUBLE    WORD 33
TA2           DOUBLE    WORD 43        TA3          DOUBLE    WORD 49
THIRD.LIFE    DOUBLE    WORD 47
```

```
EVENT DECISION.ON.MOD
''
'' DECISION IS MADE TO IMPLEMENT A STRUCTURAL MODIFICATION WHENEVER
''     RCPH(REPAIR COST PER HOUR) + ICPH(INSPECTION COST PER HOUR)
''     IS GREATER THAN MCPH(MODIFICATION COST PER HOUR)
''
       DEFINE NPDL AS AN INTEGER VARIABLE
       LET DEC.ON.MOD.SCH = "NO"
       LET TOOLING = 1ST.TOOLING
       LET MD.COST = 1ST.MD.COST
     IF PREVIOUSLY.MODIFIED = "YES"
       LET TOOLING = AD.TOOLING
       LET MD.COST = AD.MD.COST
     REGARDLESS
       FOR EVERY AIRCRAFT IN ACTIVE.FLEET
       DO
       LET MREH = ENTRY.TIME + USAGE.LIFE - TIME.V - LEAD.TIME - AHCD(LIL)/2
     IF MREH GT 0.0
       LET POST.MOD.HRS = POST.MOD.HRS + MREH
     REGARDLESS
''
     IF TMOD.PENDING(TAIL.ID) = "NO" AND SMOD.PENDING(TAIL.ID) = "NO"
      IF OCCUR.MOD(TAIL.ID) GT 0.0
       LET ACCUMULATED.HRS = ACCUMULATED.HRS + TIME.V - OCCUR.MOD(TAIL.ID)
       JUMP AHEAD
      OTHERWISE
       LET ACCUMULATED.HRS = ACCUMULATED.HRS + TIME.V - ENTRY.TIME
     HERE
     REGARDLESS
       LOOP
       LET ACCUMULATED.HRS = ACCUMULATED.HRS + AIRWAVE.TIME
       LET NPDL = TRUNC.F(LEAD.TIME/PRODUCTION.TIME)
     IF NPDL GT SIZE.OF.FLEET - IDCK
       LET NPDL = SIZE.OF.FLEET - IDCK
     REGARDLESS
       LET POST.MOD.HRS = POST.MOD.HRS+NPDL*(USAGE.LIFE-(LEAD.TIME+AHCD(LIL))/2)
       + (SIZE.OF.FLEET - IDCK - NPDL) * USAGE.LIFE
       LET ICPH = CABCD(LIL) * (1/ABCD(LIL) - 1/1ABCD(LIL)) + CAHCD(LFL) *
       (1/AHCD(LEL) - 1/1AHCD(LFL))
     IF LIL EQ LEL
       LET ICPH = CABCD(LFL) * (1/AHCD(LFL) - 1/1AHCD(LEL))
     REGARDLESS
     IF POST.MOD.HRS = 0.0
       RETURN
     OTHERWISE
       LET MCPH = (MD.COST*(IDCK+NPDL-1.NUM.OF.RETIRE-2.NUM.OF.CRASH) + TOOLING)/
       POST.MOD.HRS
       LET RCPH = COST.OF.REPAIRS / ACCUMULATED.HRS
     IF RCPH + ICPH GT MCPH
       LET SFIS = (SIZE.OF.FLEET + 1 - TRUNC.F(PRCHG/PRODUCTION.TIME)) *
       2.PRODUCTION.TIME
      IF TIME.V + LEAD.TIME LT BEGIN.PRODUCTION + PRCHG + SFIS + USAGE.LIFE
       SCHEDULE AN IMPLEMENT.MODIFICATION AT TIME.V + LEAD.TIME
```

```
LINE  CACI SIMSCRIPT II.5  RELEASE 8F                               09/01/76  PAGE  83

  54         ALWAYS
  55          LET TYP.MOD.SCH = "YES"
  56          LET PREVIOUSLY.MODIFIED = "YES"
  57         IF TES.FAILURE = "YES" AND TIME.V LE START.TEST + ACTUAL.AVG.FAT.LIFE /
  58          TEST.ACCEL.FACT
  59          LET TES.FAILURE = "NO"
  60          CANCEL THE T.IMPLEMENT.MOD
  61          DESTROY THE T.IMPLEMENT.MOD
  62         REGARDLESS
  63        REGARDLESS
  64          RETURN
  65          END

                  LOCAL VARIABLES OF THIS ROUTINE

 ACCUMULATE DOUBLE    WORD 13        ICPH         DOUBLE    WORD 15
 L.H        INTEGER   WORD  7        MCPH         DOUBLE    WORD 17
 MD.COST    DOUBLE    WORD  5        MREH         DOUBLE    WORD  9
 NETS       DOUBLE    WORD 21        NPCL         INTEGER   WORD  1
 POST.MOD.H DOUBLE    WORD 11        RCPH         DOUBLE    WORD 19
 TOOLING    DOUBLE    WORD  3
```

```
EVENT IMPLEMENT.MODIFICATION
''
'' REPRESENTS DEVELOPMENT OF MODIFICATION BECAUSE OF SERVICE EXPERIENCE
''
      LET NMD = NMD + 1
      LET FDCK = 0
      LET TO.BE.MODIFIED = IDCK
      LET BEEN.MODIFIED = 0
      LET IMP.MOD.SCH = "NO"
      LET ABCD(LEL) = 1ABCD(LEL)
      LET ABCD(LIL) = 1ABCD(LIL)

     IF NICHG GT 30
      JUMP AHEAD
    OTHERWISE
      LET SC(NICHG) = ABCD(3)
      LET SU(NICHG) = ABCD(4)
    HERE
      LET COST.OF.REPAIRS = 0.0
      LET AIRFRAME.TIME = 0.0
    IF MOD.TESTED = "NO"
      LET NMU = MU.R + ((1.0-MU.R)*.15)
      LET NSIG = SIG.R + .05
      CALL REAL.LIFE(NMU,NSIG,PREDICTED.LIFE) YIELDING ACTUAL.AVG.FAT.LIFE
      JUMP AHEAD
    OTHERWISE
      LET ACTUAL.AVG.FAT.LIFE = PREDICTED.LIFE
    HERE
      FOR EVERY AIRCRAFT IN ACTIVE.FLEET
      DO
      LET SMOD.PENDING(TAIL.ID) = "YES"
      LOOP
      RETURN
      END


                LOCAL VARIABLES OF THIS ROUTINE

L.4          INTEGER   WORD  5        NMU          DOUBLE    WORD  1
NSIG         DOUBLE    WORD  3
```

```
ROUTINE DISPLAY.OUTPUT
      DEFINE RNGE AS AN INTEGER VARIABLE
      START NEW PAGE
      WRITE MODEL(1), MODEL(2) AS S 50, "AIRCRAFT TYPE: ", 2 A 4
      SKIP 2 OUTPUT LINES
      PRINT 1 DOUBLE LINE WITH SIZE.OF.FLEET, USAGE.LIFE AS FOLLOWS
          NUMBER OF AIRCRAFT IN FLEET: *****                        AIRCRAFT SERVICE LIFE: ******* HOURS
      SKIP 1 OUTPUT LINE
      WRITE ELEMENT(1), ELEMENT(2), ELEMENT(3), ELEMENT(4)
      AS S 45, "STRUCTURAL ELEMENT: ", 4 A 4
      START NEW LINE
      SKIP 1 OUTPUT LINE
    IF PREDICTED.LIFE = 0.0
      PRINT 1 DOUBLE LINE WITH IAAFL AS FOLLOWS
   ELEMENT DESIGN FATIGUE TESTED PRIOR TO PRODUCTION                  ACTUAL AVERAGE FATIGUE LIFE: ******* HOURS
      JUMP AHEAD
    ELSE
      PRINT 1 DOUBLE LINE WITH PREDICTED.LIFE, IAAFL AS FOLLOWS
        PREDICTED AVERAGE FATIGUE LIFE: ****** HOURS                 ACTUAL AVERAGE FATIGUE LIFE: ******* HOURS
    HERE
      SKIP 2 OUTPUT LINES
      PRINT 1 DOUBLE LINE AS FOLLOWS
                              NUMBER AND TIME TO INITIATION OF AIRCRAFT DEFECTS
      SKIP 1 OUTPUT LINE
    IF NICR GT 0         LET PNICR = MICR         LET PXICR = XICR        ALWAYS
    IF NCOR GT 0         LET PNICO = MICO         LET PXICO = XICO        ALWAYS
    IF NSDM GT 0         LET PNISD = MISD         LET PXISD = XISD        ALWAYS
      PRINT 6 DOUBLE LINES WITH NICR, NCOR, NSDM, OPD, PNICR, PNICO, PNISD,
      PXICR, PXICO, PXISD, MICR, MICO, MISD AS FOLLOWS
                          FIRST CRACK          CORROSION          SERVICE DAMAGE          [illegible] DEFECTS
                          -----------          ---------          --------------          ------------------
   OCCURRENCES               *****               *****                *****                     *****
   MIN(HRS)                 ******              ******               ******                     -----
   MAX(HRS)                 ******              ******               ******                     -----
   AVG(HRS)                 ******              ******               ******                     -----
      SKIP 2 OUTPUT LINES
      PRINT 1 DOUBLE LINE AS FOLLOWS
                         NUMBER AND LENGTH OF CRACKS DETECTED AT EACH LEVEL OF INSPECTION:
      SKIP 1 OUTPUT LINE
    IF NICR(1) GT 0      LET PNACR = MACR         LET PXACR = XACR        ALWAYS
    IF NICR(2) GT 0      LET PNBCR = MBCR         LET PXBCR = XBCR        ALWAYS
    IF NICR(3) GT 0      LET PNCCR = MCCR         LET PXCCR = XCCR        ALWAYS
    IF NICR(4) GT 0      LET PNDCR = MDCR         LET PXDCR = XDCR        ALWAYS
    IF NSCR GT 0         LET PNSCR = MSCR         LET PXSCR = XSCR        ALWAYS
      PRINT 6 DOUBLE LINES WITH NICR(1), NICR(2), NICR(3), NICR(4), NSCR, PNACR,
      PNBCR, PNCCR, PNDCR, PNSCR, PXACR, PXBCR, PXCCR, PXDCR, PXSCR, MACR, MBCR,
      MCCR, MDCR, MSCR AS FOLLOWS
                          A-LEVEL          B-LEVEL          C-LEVEL          D-LEVEL          [illegible]
                          -------          -------          -------          -------          -------
   OCCURRENCES             *****            *****            *****            *****            *****
   MIN(IN)                ***.**           ***.**           ***.**           ***.**           ***.**
   MAX(IN)                ***.**           ***.**           ***.**           ***.**           ***.**
   AVG(IN)                ***.**           ***.**           ***.**           ***.**           ***.**
```

```
LINE  CACI SIMSCRIPT II.5  RELEASE 8F
                                                                     09/01/76  PAGE  86
          SKIP 2 OUTPUT LINES
          PRINT 1 DOUBLE LINE AS FOLLOWS
                              NUMBER AND AREA OF CORROSION DEFECTS DETECTED AT EACH LEVEL OF INSPECTION
          SKIP 1 OUTPUT LINE
        IF DICO(1) GT 0        LET PNACA = NACA             LET PXACA = XACA        ALWAYS
        IF DICO(2) GT 0        LET PNBCA = NBCA             LET PXBCA = XBCA        ALWAYS
        IF DICO(3) GT 0        LET PNCCA = NCCA             LET PXCCA = XCCA        ALWAYS
        IF DICO(4) GT 0        LET PNDCA = NDCA             LET PXDCA = XDCA        ALWAYS
        IF DSCO GT 0           LET PNSCA = NSCA             LET PXSCA = XSCA        ALWAYS
          PRINT 6 DOUBLE LINES WITH DICO(1), DICO(2), DICO(3), DICO(4), DSCO, PNACA,
          PNBCA, PNCCA, PNDCA, PNSCA, PXACA, PXBCA, PXCCA, PXDCA, PXSCA, MACA, MBCA,
          MCCA, MDCA, MSCA AS FOLLOWS
                               A-LEVEL            B-LEVEL             C-LEVEL            D-LEVEL            SPECIAL
                               -------            -------             -------            -------            -------
      OCCURRENCES               *****              *****               *****              *****              *****
      MIN(SQ.IN)               ***.**             ***.**              ***.**             ***.**             ***.**
      MAX(SQ.IN)               ***.**             ***.**              ***.**             ***.**             ***.**
      AVG(SQ.IN)               ***.**             ***.**              ***.**             ***.**             ***.**
         SKIP 1 OUTPUT LINE
         PRINT 1 LINE AS FOLLOWS
      INSPECTION INTERVALS(HRS)
         SKIP 1 OUTPUT LINE
         PRINT 1 DOUBLE LINE WITH IAHCD(1), IAHCD(2), IAHCD(3), IAHCD(4) AS FOLLOWS
      INITIAL                   *****              *****               *****              *****
         IF NICHG GT 30       LET NICHG = 30       REGARDLESS
         FOR I = 2 TO NICHG
         DO
         PRINT 1 DOUBLE LINE WITH I, IAHCD(1), IAHCD(2), SC(I), SD(I) AS FOLLOWS
        **                      *****              *****               *****              *****
         LOOP
         SKIP 1 OUTPUT LINE
         LET SNSIC = SNSIC + NSIC
         PRINT 1 LINE WITH NSIC AS FOLLOWS
      NUMBER OF SPECIAL INSPECTIONS CONDUCTED: **
         LET SNMD = SNMD + NMD
         PRINT 1 LINE WITH NMD AS FOLLOWS
      NUMBER OF STRUCTURAL MODIFICATIONS: **
         PRINT 1 LINE WITH ACTUAL.AVG.FAT.LIFE AS FOLLOWS
      FINAL ACTUAL AVERAGE MODIFIED FATIGUE LIFE: ******* HOURS
         LET SNSMD = SNSMD + NSMD
         PRINT 1 LINE WITH NSMD AS FOLLOWS
      NUMBER OF AIRCRAFT MODIFIED IN SERVICE: *****
         SKIP 1 OUTPUT LINE
         PRINT 1 DOUBLE LINE AS FOLLOWS
                    STRUCTURAL FAILURES                                          RESIDUAL STRENGTH EQUALS FAIL-SAFE STRENGTH
         PRINT 1 DOUBLE LINE AS FOLLOWS
               AIRCRAFT NO.          FLT. HOURS                                        AIRCRAFT NO.       FLT. HOURS
         PRINT 1 DOUBLE LINE AS FOLLOWS
               ------------          ----------                                        ------------       ----------
         LET RNGE = MAX.F(NRES,NSFL)
         FOR I = 1 TO RNGE
         DO
        IF NSFL GE I AND NRES GE I
```

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

```
LINE  CACI SIMSCRIPT II.5  RELEASE 8F                                    09/01/76  PAGE  87

 73          WRITE ACID(I), FLTHR(I), APID(I), STIM(I) AS S 21, I 5, S 14, I 6, S 41,
 74          I 5, S 14, I 6
 75          JUMP AHEAD
 76         ELSE
 77    ''
 78         IF NRES GE I
 79           WRITE APID(I), STIM(I) AS S 67, I 5, S 14, I 6
 80          JUMP AHEAD
 81         ELSE
 82          WRITE ACID(I), FLTHR(I) AS S 21, I 5, S 14, I 6
 83        HERE
 84          START NEW LINE
 85          LOOP
 86          RETURN
 87          END


                LOCAL VARIABLES OF THIS ROUTINE

 I.1         INTEGER  WORD  5       I.2        INTEGER  WORD  6
 I.3         INTEGER  WORD  7       J.1        INTEGER  WORD  8
 K.1         INTEGER  WORD 10       K.2        INTEGER  WORD 11
 K.3         INTEGER  WORD 12       K.4        INTEGER  WORD 13
 L.1         INTEGER  WORD 14       L.11       INTEGER  WORD 31
 L.13        INTEGER  WORD 42       L.15       INTEGER  WORD 53
 L.17        INTEGER  WORD 54       L.19       INTEGER  WORD 75
 L.21        INTEGER  WORD 76       L.26       INTEGER  WORD 77
 L.28        INTEGER  WORD 78       L.3        INTEGER  WORD 15
 L.5         INTEGER  WORD 16       L.7        INTEGER  WORD 17
 L.9         INTEGER  WORD 18       M.1        INTEGER  WORD  9
 PNACA       DOUBLE   WORD 55       PNACR      DOUBLE   WORD 33
 PNBCA       DOUBLE   WORD 59       PNBCR      DOUBLE   WORD 37
 PNCCA       DOUBLE   WORD 63       PNCCR      DOUBLE   WORD 41
 PNDCA       DOUBLE   WORD 67       PNDCR      DOUBLE   WORD 45
 PNSCA       DOUBLE   WORD 71       PNSCR      DOUBLE   WORD 49
 PNICA       DOUBLE   WORD 23       PNICR      DOUBLE   WORD 19
 PNISA       DOUBLE   WORD 27       PXACA      DOUBLE   WORD 57
 PXACR       DOUBLE   WORD 35       PXBCA      DOUBLE   WORD 61
 PXBCR       DOUBLE   WORD 39       PXCCA      DOUBLE   WORD 65
 PXCCR       DOUBLE   WORD 43       PXDCA      DOUBLE   WORD 69
 PXDCR       DOUBLE   WORD 47       PXSCA      DOUBLE   WORD 73
 PXSCR       DOUBLE   WORD 51       PXICA      DOUBLE   WORD 25
 PXICR       DOUBLE   WORD 21       PXISA      DOUBLE   WORD 29
 R.1         DOUBLE   WORD  3       RNGE       INTEGER  WORD  1
```

```
ROUTINE SUMMARY
      DEFINE RNGE AS AN INTEGER VARIABLE
      START NEW PAGE
      WRITE MODEL(1), MODEL(2) AS S 50, "AIRCRAFT TYPE: ", 2 A 4
      SKIP 2 OUTPUT LINES
      PRINT 1 DOUBLE LINE WITH SIZE.OF.FLEET, USAGE.LIFE AS FOLLOWS
              NUMBER OF AIRCRAFT IN FLEET: *****                          AIRCRAFT SERVICE LIFE: ******* HOURS
      SKIP 1 OUTPUT LINE
      WRITE SEL1, SEL2, SEL3 AS S 45, "SUMMARY OF STRUCTURAL ELEMENT: ", 2 A 4,
      A 3
      SKIP 2 OUTPUT LINES
      PRINT 1 DOUBLE LINE AS FOLLOWS
                                 NUMBER AND TIME TO INITIATION OF AIRCRAFT DEFECTS
      SKIP 1 OUTPUT LINE
     IF GOICR GT 0        LET PSICR = SICR    LET PLICR = LICR            ALWAYS
     IF GOCOR GT 0        LET PSICO = SICO    LET PLICO = LICO            ALWAYS
     IF GOSOM GT 0        LET PSISD = SISD    LET PLISD = LISD            ALWAYS
      PRINT 6 DOUBLE LINES WITH GOICR, GOCOR, GOSDM, GOPD, PSICR, PSICO, PSISD,
      PLICR, PLICO, PLISD, AICR, AICO, AISD AS FOLLOWS
                      FIRST CRACK           CORROSION            SERVICE DAMAGE           PRODUCTION DEFECTS
                      -----------           ---------            --------------           ------------------
   OCCURRENCES           *****                *****                 *****                    *****
   MIN(HRS)             ******               ******                ******                    -----
   MAX(HRS)             ******               ******                ******                    -----
   AVG(HRS)             ******               ******                ******                    -----
      SKIP 2 OUTPUT LINES
      PRINT 1 DOUBLE LINE AS FOLLOWS
                          NUMBER AND LENGTH OF CRACKS DETECTED AT EACH LEVEL OF INSPECTION
      SKIP 1 OUTPUT LINE
     IF GOICR(1) GT 0     LET PSACR = SACR    LET PLACR = LACR            ALWAYS
     IF GOICR(2) GT 0     LET PSHCR = SHCR    LET PLHCR = LHCR            ALWAYS
     IF GOICR(3) GT 0     LET PSCCR = SCCR    LET PLCCR = LCCR            ALWAYS
     IF GOICR(4) GT 0     LET PSDCR = SDCR    LET PLDCR = LDCR            ALWAYS
     IF GOSCR GT 0        LET PSSCR = SSCR    LET PLSCR = LSCR            ALWAYS
      PRINT 6 DOUBLE LINES WITH GOICR(1), GOICR(2), GOICR(3), GOICR(4), GOSCR,
      PSACR, PSHCR, PSCCR, PSDCR, PSSCR, PLACR, PLHCR, PLCCR, PLDCR, PLSCR,
      AACR, AHCR, ACCR, ADCR, ASCR AS FOLLOWS
                      A-LEVEL          H-LEVEL          C-LEVEL          D-LEVEL          SPECIAL
                      -------          -------          -------          -------          -------
   OCCURRENCES         *****            *****            *****            *****            *****
   MIN(IN)            ***.**           ***.**           ***.**           ***.**           ***.**
   MAX(IN)            ***.**           ***.**           ***.**           ***.**           ***.**
   AVG(IN)            ***.**           ***.**           ***.**           ***.**           ***.**
      SKIP 2 OUTPUT LINES
      PRINT 1 DOUBLE LINE AS FOLLOWS
                     NUMBER AND AREA OF CORROSION DEFECTS DETECTED AT EACH LEVEL OF INSPECTION
      SKIP 1 OUTPUT LINE
     IF GOICO(1) GT 0     LET PSACA = SACA    LET PLACA = LACA            ALWAYS
     IF GOICO(2) GT 0     LET PSHCA = SHCA    LET PLHCA = LHCA            ALWAYS
     IF GOICO(3) GT 0     LET PSCCA = SCCA    LET PLCCA = LCCA            ALWAYS
     IF GOICO(4) GT 0     LET PSDCA = SDCA    LET PLDCA = LDCA            ALWAYS
     IF GOSCO GT 0        LET PSSCA = SSCA    LET PLSCA = LSCA            ALWAYS
      PRINT 6 DOUBLE LINES WITH GOICO(1), GOICO(2), GOICO(3), GOICO(4), GOSCO,
LINE  CACI SIMSCRIPT II.5  RELEASE 8F                             09/01/76  PAGE  89

 38        PSACA, PSHCA, PSCCA, PSDCA, PSSCA, PLACA, PLBCA, PLCCA, PLDCA, PLSCA,
 39        AACA, AHCA, ACCA, ADCA, ASCA AS FOLLOWS
                          A-LEVEL           B-LEVEL           C-LEVEL           D-LEVEL           SPECIAL
                          -------           -------           -------           -------           -------
      OCCURRENCES          *****             *****             *****             *****             *****
      MIN(SQ.IN)          ***.**            ***.**            ***.**            ***.**            ***.**
      MAX(SQ.IN)          ***.**            ***.**            ***.**            ***.**            ***.**
      AVG(SQ.IN)          ***.**            ***.**            ***.**            ***.**            ***.**
 40           SKIP 1 OUTPUT LINE
 41        PRINT 1 LINE AS FOLLOWS
      INSPECTION INTERVALS(HRS)
 42        PRINT 1 DOUBLE LINE WITH IABCD(1), IABCD(2), IABCD(3), IABCD(4) AS FOLLOWS
      INITIAL              *****             *****             *****             *****
 43        PRINT 1 DOUBLE LINE WITH IABCD(1), IABCD(2), NCINS, NDINS AS FOLLOWS
      SHORTEST             *****             *****             *****             *****
 44        PRINT 1 DOUBLE LINE WITH IABCD(1), IABCD(2), XCINS, XDINS AS FOLLOWS
      LONGEST              *****             *****             *****             *****
 45        SKIP 1 OUTPUT LINE
 46        PRINT 1 LINE WITH SNSIC AS FOLLOWS
      NUMBER OF SPECIAL INSPECTIONS CONDUCTED: ***
 47        PRINT 1 LINE WITH SNMD AS FOLLOWS
      NUMBER OF STRUCTURAL MODIFICATIONS: ***
 48        PRINT 1 LINE WITH SNSMD AS FOLLOWS
      NUMBER OF AIRCRAFT MODIFIED IN SERVICE: *****
 49        SKIP 1 OUTPUT LINE
 50        PRINT 1 DOUBLE LINE AS FOLLOWS
                    STRUCTURAL FAILURES                                      RESIDUAL STRENGTH EQUALS FAIL-SAFE STRENGTH
 51        PRINT 1 DOUBLE LINE AS FOLLOWS
      AIRCRAFT NO.        FLT. HOURS        STA. NO.                         AIRCRAFT NO.        FLT. HOURS        STA. NO.
 52        PRINT 1 DOUBLE LINE AS FOLLOWS
      ------------        ----------        --------                         ------------        ----------        --------
 53        LET RNGE = MAX.F(SNRES,SNSFL)
 54        FOR I = 1 TO RNGE
 55        DO
 56       IF SNSFL GE I AND SNRES GE I
 57        WRITE SACID(I), SFLTHR(I), SSTAN(I), SAPID(I), SSTIX(I), SFLNR(I) AS S 13,
 58        I 5, S 14, I 6, S 12, A 4, S 24, I 5, S 14, I 6, S 13, A 4
 59        JUMP AHEAD
 60       ELSE
 61  ''
 62       IF SNRES GE I
 63        WRITE SACID(I), SSTIM(I), SFLNR(I) AS S 76, I 5, S 14, I 6, S 13, A 4
 64        JUMP AHEAD
 65       ELSE
 66        WRITE SACID(I), SFLTHR(I), SSTAN(I) AS S 13, I 5, S 14, I 6, S 12, A 4
 67      HERE
 68        START NEW LINE
 69        LOOP
 70        RETURN
 71        END
```

```
LINE  CACI SIMSCRIPT II.5  RELEASE 8F                                 09/01/76  PAGE  90

                     LOCAL VARIABLES OF THIS ROUTINE

 I.1          INTEGER   WORD  5        I.2          INTEGER  WORD  6
 I.3          INTEGER   WORD  7        J.1          INTEGER  WORD  8
 K.1          INTEGER   WORD 10        K.2          INTEGER  WORD 11
 K.3          INTEGER   WORD 12        K.4          INTEGER  WORD 13
 L.1          INTEGER   WORD 14        L.11         INTEGER  WORD 32
 L.13         INTEGER   WORD 53        L.15         INTEGER  WORD 54
 L.17         INTEGER   WORD 75        L.19         INTEGER  WORD 76
 L.21         INTEGER   WORD 77        L.3          INTEGER  WORD 15
 L.5          INTEGER   WORD 16        L.7          INTEGER  WORD 17
 L.9          INTEGER   WORD 31        M.1          INTEGER  WORD  9
 PLACA        DOUBLE    WORD 57        PLACB        DOUBLE   WORD 35
 PLBCA        DOUBLE    WORD 61        PLBCB        DOUBLE   WORD 39
 PLCCA        DOUBLE    WORD 65        PLCCB        DOUBLE   WORD 43
 PLDCA        DOUBLE    WORD 69        PLDCB        DOUBLE   WORD 47
 PLSCA        DOUBLE    WORD 73        PLSCB        DOUBLE   WORD 51
 PLICO        DOUBLE    WORD 25        PLICB        DOUBLE   WORD 21
 PLISD        DOUBLE    WORD 29        PSACA        DOUBLE   WORD 55
 PSACB        DOUBLE    WORD 33        PSBCA        DOUBLE   WORD 59
 PSBCB        DOUBLE    WORD 37        PSCCA        DOUBLE   WORD 63
 PSCCB        DOUBLE    WORD 41        PSDCA        DOUBLE   WORD 67
 PSDCB        DOUBLE    WORD 45        PSSCA        DOUBLE   WORD 71
 PSSCB        DOUBLE    WORD 49        PSICO        DOUBLE   WORD 23
 PSICB        DOUBLE    WORD 19        PSISD        DOUBLE   WORD 27
 R.1          DOUBLE    WORD  3        PAGE         INTEGER  WORD  1
```